



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA DE SOFTWARE

Árbol Genealógico de la Informática

Informatics Genealogy Project

Realizado por
Francisco Javier Pérez Jorge

Tutorizado por
Antonio Vallecillo Moreno
Lola Burgueño Caballero

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JUNIO DE 2018

Fecha defensa:

Fdo. El/la Secretario/a del Tribunal



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA

Resumen

La American Mathematical Society mantiene desde hace 21 años el proyecto, <http://www.genealogy.ams.org/>, que permite conocer el “árbol genealógico” de las matemáticas a nivel internacional, en la que muestra información sobre la relación entre los matemáticos y sus directores de tesis doctoral (padres académicos) y sus doctorandos (hijos académicos). Este es un proyecto que ha ido creciendo y popularizando con el tiempo (véase por ejemplo el reciente artículo sobre el proyecto en las Notas de la AMS: <http://www.ams.org/publications/journals/notices/201705/rnoti-p466.pdf>).

En presente trabajo de fin de grado se ha diseñado y desarrollado una aplicación Web que tenga la misma funcionalidad, pero centrándose en el campo de la Informática. Como distintivo, se pretende desarrollar con el nuevo lenguaje de programación Go (<https://golang.org/>), que está adquiriendo cada vez más protagonismo. El trabajo de fin de grado va a permitir conocer ese lenguaje y explorar las posibilidades que ofrece para el desarrollo de aplicaciones Web, en primera persona.

Este proyecto ha sido una iniciativa de la Sociedad Nacional de Ingeniería del Software y Tecnologías de Desarrollo de Software (SISTEDES), que tiene interés en los resultados del proyecto en caso de culminar con éxito.

Palabras clave

Golang, Postgres, Aplicación web, Árbol genealógico, Informática

Abstract

The American Mathematical Society started a project (<http://www.genealogy.ams.org/>) 21 years ago with the aim of building a genealogy tree about mathematicians from all over the world, showing the information about the relationships between PhD students and their supervisors. This project has become popular and currently it is widely used by the mathematicians. (For example, the following article describes this project <http://www.ams.org/publications/journals/notices/201705/rnoti-p466.pdf>).

In this final project, we have designed and developed a Web application with similar functionality but focusing on the computer science area. Furthermore, this project has been developed using the programming language “Go” (<https://golang.org/>), which is gaining popularity.

This project is an initiative of the SISTEDES society (Sociedad Nacional de Ingenieria del Software y Tecnologías del Desarrollo de Software), which has special interest in the results of the project.

Keywords

Golang, Postgres, Web Application, Geanology tree, Informatics



Índice

1	Introducción.....	9
1.1	Objetivo.....	9
1.2	Contenido y estructura del documento	10
2	Motivación del proyecto	13
2.1	Objeto	14
2.2	Antecedentes.....	14
2.3	Descripción de la situación actual.....	14
2.3	Metodología y fases del trabajo	15
3	Requisitos iniciales	17
4	Alcance	21
4.1.	Identificación de partes.....	21
4.2.	Plan de gestión de alcance	22
4.3	Suposiciones preliminares.....	23
4.4.	Criterios de aceptación.....	23
4.5.	Entorno de despliegue	24
4.6.	Entorno de desarrollo	25
4.7.	Software de terceros.....	25
5	Análisis de requisitos.....	27
5.1.	Requisitos funcionales	27
5.2.	Requisitos no funcionales	33
5.3.	Requisitos de documentación	36
6	Análisis y gestión de riesgos	39
7	Modelado y diseño.....	43
7.1.	Casos de uso	44
7.2.	Diagrama de clases	46
7.3.	Modelo de navegación.....	48
7.4.	Diagramas de secuencia	50

8 Implementación	61
8.1. Estructura general.....	61
8.2. Tecnologías fundamentales.....	62
8.3. Tecnologías adicionales.....	63
8.4. Despliegue.....	64
9 Mantenimiento y pruebas	67
9.1. Consideraciones previas.....	68
9.2. Test Experimental	68
9.3. Perfil de los encuestados.....	74
9.4. Resultados del Test	74
10 Conclusiones	77
10.1. Objetivos cumplidos.....	77
10.2. Dificultades encontradas	78
10.3. Líneas futuras.....	79
Referencias.....	81
Manual de Usuario	83
Normas aplicadas	89

1

Introducción

Comenzamos este documento y en especial en este capítulo con una breve introducción del proyecto que ha motivado realizar este trabajo de fin de grado que puede servir como referencia a más personas.

Previo a comenzar con la memoria, se empezará exponiendo el objetivo que tiene esta memoria y continuará con la estructura seguida en la realización de este documento en donde se explicará todo detalle de la realización de este proyecto, “Árbol genealógico de la informática”.

1.1 Objetivo

Con este documento se presenta al lector de forma estructurada las distintas fases llevadas a cabo en la elaboración de este proyecto.

Primeramente, trataremos de explicar los objetivos que tratamos de buscar con la realización de este proyecto, seguido de la metodología que hemos seguido para su realización.

A continuación, se desarrollará todo el trabajo realizado para la elaboración de este proyecto. Así como, la definición de las distintas etapas llevadas a cabo para seguir con la planificación del proyecto.

Además, se incluirán en distintos puntos valoraciones personales realizadas sobre la tecnología escogida para que pueda servir de referencia más personas.

Finalmente, se incluirán las conclusiones respectivas al proyecto realizado, valorando todos los puntos realizados y enfocándonos en la proyección de la aplicación para la actualidad y para el futuro.

1.2 Contenido y estructura del documento

-Introducción del proyecto: Donde se comentarán de las motivaciones que llevaron a desarrollar este proyecto, un resumen de lo que es este proyecto en sí y los objetivos que se esperan cumplir.

-Alcance y participantes: Estudio del público objetivo para utilizar la aplicación y definición de los distintos roles utilizados en el sistema junto con su funcionamiento.

-Requisitos: Definición de los requisitos funcionales y no funcionales y análisis de los requisitos definidos que deberá cumplir nuestro sistema junto una clasificación por su prioridad de importancia para cumplir los objetivos del proyecto

-Requisitos no funcionales: Definición de los requisitos que deberá cumplir nuestro proyecto con el fin de asegurar unos estándares de calidad mínimos en el resultado final.

-Análisis de riesgos: Definición de los posibles eventos que se produzcan que puedan poner en riesgo el desarrollo y/o funcionamiento del proyecto junto con una clasificación según la probabilidad de que ocurra y su impacto sobre el proyecto.

-Modelado del sistema: Inclusión de un modelado de distribución del sistema y de navegación que ayudará a comprender el funcionamiento de nuestra aplicación web.

-Metodologías y herramientas: Definición de las metodologías empleadas para el desarrollo del proyecto, así como de las tecnologías utilizadas con su argumentación incluida.

-Mantenimiento: Definición del plan establecido para preservar el correcto mantenimiento del proyecto, así como permitir posibles modificaciones en un futuro.

-Pruebas: Definición de las pruebas llevadas a cabo con el fin de comprobar y asegurar el correcto funcionamiento del sistema y su nivel de seguridad.



2

Motivación de proyecto

La idea de este proyecto web nace de la proposición de los directores y sus posteriores acuerdos con el autor de trabajo de fin de grado.

La necesidad de conseguir simular un sistema ya existente ponía en el centro una idea clara que solo debía ser retocada dada a las funcionalidades que se querían añadir sobre la original. Estas ideas fueron consensuándose mutuamente hasta llegar a la especificación final que incluiría una interfaz web accesible por todas las personas con acceso a internet.

Finalmente, este desarrollo contó con la supervisión y ayuda de los directores del proyecto que permitieron resolver dudas y dar consejos sobre las posibles elecciones.

2.1 Objeto

“Árbol Genealógico de la Informática” a desarrollar consiste en una aplicación web multiplataforma con el fin de conseguir ser una herramienta que permitirá servir de entretenimiento a todas las personas que tengan curiosidad sobre las relaciones formadas mediante las tesis doctorales del campo de la informática.

Esta funcionalidad, incluye la posibilidad de visualizar información sobre doctores y sus tesis relacionadas con las propias instituciones donde se llevaron a cabo y así relacionarlas con los directores y el tribunal que participó en su lectura. En sí, realizada en una aplicación web de acceso universal y gratuito sería accesible por todas las personas que tengan interés.

2.2 Antecedentes

Como hemos dicho anteriormente, el dominio web <http://www.genealogy.ams.org/>, ha experimentado un crecimiento notable últimamente en términos de popularidad. Es por eso, que, ante semejante auge apoyado por el innegable crecimiento del tráfico web y la inexistencia de una herramienta destinada a un sector suficientemente grande, se haya decidido intentar crear un nuevo sistema, el cual, con algunas innovaciones presente un espacio para los doctores del ámbito informático.

2.3 Descripción de la situación actual

Una vez comentada la situación que produjo la creación de la idea y posterior proyecto, se pasó a un proceso de diseño. En este proceso, los directores del Trabajo de fin de grado, mediante una serie de reuniones con el autor de este, acordaron las funcionalidades y las fases de desarrollo a realizar que contendría el sistema.

El desarrollo a partir de la definición de requisitos inicial se ha desarrollado a través de una metodología iterativa que nos ha permitido a la vez que

incrementar requisitos, ir realizando varias partes del proyecto que nos servirían posteriormente. De este modo, hemos conseguido pasar de un prototipo funcional hacia una aplicación web totalmente operativa.

Para concluir su desarrollo, el sistema se vio envuelto en una fase de análisis de uso ejecutado por distintos usuarios que han valorado funcionalidad de uso de la herramienta y distintos aspectos relativa a ella que nos ha ayudado a mejorar notablemente su resultado.

2.3 Metodología y fases del trabajo

Debido a que este proyecto consistía en la creación de una aplicación web, aunque con unas funcionalidades muy claras, una tecnología innovadora desconocida tanto para los directores del trabajo como para el autor, se decidió seguir una metodología ágil que nos permitiera acometer el desarrollo del proyecto de una manera más permisiva a modificaciones en caso de encontrarnos con limitaciones debido a las tecnologías.

Ante esto, cabe decir que no se siguió literalmente ninguna metodología específica, pero realizamos un desarrollo iterativo en el que, en cada reunión con los clientes, en este caso los directores del trabajo, se llevaba una parte del desarrollo realizado y se debatían las funcionalidades que faltaban y la siguiente reunión para debatir su implementación.

A pesar de no haber seguido una metodología explícita, se definen unos roles básicos para la realización del proyecto en función de los participantes de esta. Estos han sido:

- **Jefe de Proyecto:** Desempeñado por el autor del trabajo de fin de grado, ha realizado las tareas de jefe de proyecto en las que reuniéndose con los clientes del proyecto y escuchando sus peticiones ha definido los requisitos del proyecto y ha tomado las decisiones oportunas sobre este.

- **Desarrollador:** Desempeñado por el autor del trabajo de fin de grado, ha realizado todas las tareas de implementación y desarrollo acordadas entre el jefe de proyecto y el cliente.
- **Cliente:** Desempeñado por los directores del proyecto, se han encargado de interactuar con el jefe de proyecto mediante reuniones presentándoles sus ideas sobre el resultado final que querían y opinando sobre el desarrollo actual. Esto ha permitido en parte una metodología iterativa.

Gracias a estos roles, se ha podido desarrollar un proyecto en el que cada iteración ha consistido en estas fases:

- **Recolección de requisitos:** efectuada mediante una reunión, se acordaban las siguientes funcionalidades a implementar en el sistema.
- **Diseño:** se preparaba un diseño que permitiría al desarrollador tener una idea de cómo se implementarían las nuevas funcionalidades.
- **Implementación:** Guiado por los modelos creados, desarrollaba el código pertinente.
- **Pruebas:** Última fase en la que se probaba el funcionamiento de las actualizaciones en el sistema

3

Requisitos iniciales

El proceso de recolección de requisitos en sí es una de las partes más fundamentales del propio desarrollo y la que afectará en mayor medida al resultado del proyecto en caso de haberse realizado una tarea de forma errónea o ineficiente. Este impacto negativo se verá afectado innegablemente en los recursos económicos o de tiempo primordialmente ya que la falta de inclusión de un requisito funcional provoca una modificación de la documentación, y un análisis de la modificación que se deberá realizar sobre el código con todo lo que ello conlleva.

Los requisitos iniciales, o bien, los requisitos que se acuerdan en las primeras reuniones pueden carecer de profundidad sobre la funcionalidad total del proyecto, pero en una primera fase en la que se trabaja en un nivel de abstracción muy alto, es muy útil disponer de unas ideas básicas y claras sobre el objetivo del proyecto para comenzar su planificación de tecnologías y recursos utilizados.

A continuación, en la tabla 1 se muestran los requisitos básicos que se acordaron en la primera reunión del trabajo de fin de grado en el que se presentó la idea de proyecto y se accedió a hacerlo.

Requisitos	Descripción
Sistema accesible mediante internet	Se describiría el objetivo del proyecto como el desarrollo de un sistema accesible universalmente mediante el uso de conexiones a internet.
Aplicación web	Mas como un requisito derivado del anterior, el sistema se conformaría como una aplicación web.
Alojamiento	La aplicación web deberá ser alojada en algún dominio para su uso a través de internet.
Usar un sistema gestor de base de datos	Dado a que la funcionalidad del sistema incluiría la manipulación de datos, será necesario contar con un sistema gestor de bases de datos apropiado al flujo de información con el que contaremos
Mostrar información sobre Tesis doctorales y Doctores	El sistema mostrará información sobre tesis doctorales pertenecientes al campo de la informática y sobre los doctores que las defendieron o participaron activamente tanto en el desarrollo de la tesis como en su lectura
Mostrar relaciones entre doctores	Mostrar de forma explícita las relaciones generadas con el desarrollo de una tesis doctoral entre la dirección del proyecto y el alumno.
Permitir la navegabilidad entre doctores	Se potenciará el uso del concepto de las relaciones entre directores y

	<p>alumno para permitir que también puedas visualizar la relaciones sobre los antecesores de los directores conformando un árbol genealógico de la ingeniería</p>
--	---

*Tabla 1: Requisitos iniciales



4

Alcance

A la hora de realizar un proyecto, también es imprescindible saber imponer unos límites sobre el proyecto llevado a cabo y las distintas sociedades, productos y tecnologías que tengan una relación con este.

4.1. Identificación de partes

Principalmente, se deberán definir con exactitud las partes implicadas en la creación del proyecto ya que debe haber constancia de que persona u organismo tendrán responsabilidad en caso de cualquier incidencia. En este caso, también es necesario definir al cliente ya que cualquier posible modificación sobre su idea o estado económico podría influir activamente sobre el proyecto.

-Proyecto:

-Título: Árbol Genealógico de la Informática

-Id: 001

-Jefe de proyecto:

-Nombre y apellidos: Francisco Javier Pérez Jorge

-Email: francis_95@hotmail.es

-Clientes:

-Nombre y apellidos: Antonio Vallecillo Moreno

-Email: av@lcc.uma.es

-Nombre y apellidos: Loli Burgueño Caballero

-Email: loli@lcc.uma.es

4.2. Plan de gestión de alcance

El plan de gestión del alcance trata de definir los criterios que se deben seguir en la elaboración, aceptación y actualización de distintos procesos referentes a la definición del alcance. Principalmente podemos resumir el plan de gestión de alcance que se debería realizar de la siguiente forma:

-Preparar la declaración del alcance: Trata de una declaración preliminar que dará las bases para una extensión posterior y deberá ser revisada por los interesados claves del proyecto.

-Crear la Estructura de descomposición del trabajo, EDT: Todo el equipo del proyecto trabajara en la elaboración de la EDT en la que se incluirán todas las actividades necesarias para poder finalizar el proyecto.

-Verificar que los entregables estén completos: Se definirá un proceso que trate de confirmar que cada entregable está completo.

- Gestionando las peticiones de cambios a la declaración del alcance: Todas las peticiones de modificaciones significativas que tengan un impacto sobre la planificación principal deberán seguir un proceso de control de cambios.

4.3 Suposiciones preliminares

En este apartado, se tratan de incluir todos los conceptos que se dan por supuestos en la definición del sistema.

La importancia de dejar por escrito escritas estas suposiciones reside en que posibles modificaciones externas a nuestro sistema pueden terminar afectando a la propia funcionalidad del sistema por lo que se debe tener constancia de todos los puntos en los que se confían por hechos.

-Los usuarios insertarán nueva información ayudándonos a ir incrementando el valor de la aplicación y del sistema.

-Toda la información mostrada es real y esto será confirmada por los administradores a la hora de aceptar la inserción de datos.

-Al estar alojada la aplicación web en internet, los usuarios deberán disponer de conexión a internet si quieren acceder.

4.4. Criterios de aceptación

A la hora de poder dar por terminado un proyecto, se necesitan definir con anterioridad, que criterios objetivos servirán para poder dar como exitosa la finalización el producto resultante. Estos criterios son de vital importancia para evitar posibles desencuentros en la entrega del producto que pueda desencadenar en una pérdida de más recursos o que llegue a acabar por la vía judicial.

Por consiguiente, en la realización de este proyecto se acordaron como criterios justos de aceptación los siguientes:

-El proyecto en su conjunto debería finalizarse antes del plazo máximo de entrega (29 de junio de 2018)

-El producto resultante debería contener todas las funcionalidades

pedidas previamente en las reuniones con los clientes. Estas funcionalidades, corresponden con los requisitos funcionales que se acordaron para su implementación.

-El producto resultante podrá ser visitado por todo el mundo que así lo requiera para comprobar su funcionalidad.

-El análisis deberá haber cumplido un conjunto de pruebas sobre el código que confirmen la correcta funcionalidad del sistema.

4.5. Entorno de despliegue

Debido a las singularidades de este proyecto sobre la especificación predeterminada de la tecnología que se iba a utilizar en un modo de investigación y análisis, la elección del entorno de despliegue se dejó a libre elección por parte del autor del trabajo de fin de grado.

Tras un análisis exhaustivo de las posibles opciones de despliegue y tras comprobar la incompatibilidad de varias herramientas que ofrecen un servicio de alojamiento web con la tecnología que nosotros hemos usado (Go), se optó por simular el sistema de desarrollo en una máquina virtual alojada en la nube en la que desplegar nuestra aplicación modo local para así poder acceder a ella a través de la dirección IP.

Las ventajas que nos brindan esta elección es la de que no necesitábamos un PaaS (Plataforma como servicio) que fuera compatible con nuestra versión de Golang, sino que simulando nuestro entorno completo podemos ejecutarla sin ningún problema y en un sistema totalmente abierto a posibles actualizaciones de las versiones.

4.6. Entorno de desarrollo

El entorno de desarrollo para este proyecto ha sido el ordenador personal del autor del trabajo de fin de grado el cual cuenta con las siguientes especificaciones de software:

- Sistema Operativo: Ubuntu (Linux) 16.04 LTS
- Editor de texto: Sublime Text 3
- Software instalado:
- Golang 1.10.1
- Postgres 1.15

4.7. Software de terceros

Durante el desarrollo de un software propio, es usual utilizar librerías y frameworks de terceros que ayudan a la propia programación permitiendo tener un código mucho más limpio y estructurado. Pero, el uso de estas librerías debe realizarse con especial cuidado ya que al insertar software de terceros en tu propio sistema estás incluyendo también posibles malas prácticas de programación de los desarrolladores o vulnerabilidades de su código. Es por eso por lo que se aconseja en todo momento un uso responsable de las librerías utilizando versiones estables y testadas y de fuentes originales.

En este caso, las librerías utilizadas son:

- jQuery
- Vis.js
- Bootstrap 4
- Datatable JS
- Golang Cors
- HTTP/r Go



Análisis de requisitos

Una de las fases más importantes de la documentación y del desarrollo de un proyecto software es el análisis y especificación de requisitos.

El análisis de requisitos consiste en el proceso de definir y documentar las funcionalidades que debe cumplir el proyecto. El correcto desarrollo del software depende de la correcta y exacta descripción de los requisitos del sistema. Definir los requisitos y una definición exacta y correcta es la principal tarea de la ingeniería de requisitos (Zhi Jin, Environment Modeling-Based Requirements Engineering for Software Intensive Systems, 2017)

5.1. Requisitos funcionales

Estos requisitos se corresponden con las necesidades específicas del cliente recolectados mediante las reuniones con el propio cliente.

Estos requisitos pueden verse ligados estrechamente con los casos de uso del sistema y es que, desde un aspecto más lógico, los requisitos funcionales se podrían deducir lógicamente de la segmentación de los casos de uso.

A continuación, procederemos a exponer todos los casos de uso definidos en nuestro sistema:

-Visualizar en una pestaña información sobre doctores.

-Identificador: RF-001

-Tipo: Requisito funcional

-Descripción: Deberá existir una pestaña en la que podamos ver la información personal básica de un doctor junto información sobre los directores de sus tesis doctorales, el tribunal de sus tesis doctorales y sus alumnos de doctorado desde donde podremos navegar hacia la página de estos.

-Prioridad: Alta

-Derivados:

-Guardar información sobre doctores

-Identificador: RF-001-D1

-Tipo: Requisito funcional

-Descripción: Deberemos disponer de información sobre doctores en nuestro sistema

-Prioridad: Alta

-Guardar información sobre las relaciones entre doctores

-Identificador: RF-001-D2

-Tipo: Requisito funcional

-Descripción: Deberemos disponer de información sobre las relaciones entre doctores producidas a través de las tesis doctorales en nuestro sistema

-Prioridad: Alta

-Navegar entre doctores

-Identificador: RF-001-D3

-Tipo: Requisito funcional

-Descripción: Mediante la representación utilizada para mostrar la información de las relaciones entre doctores, se podrá navegar hacia la página personal de ese doctor que incluirá su información si esa está en nuestro sistema

-Prioridad: Alta

-Visualizar en una pestaña información sobre tesis doctorales

-Identificador: RF-002

-Tipo: Requisito funcional

-Descripción: Deberá existir una página para mostrar información sobre tesis doctorales de modo que exista información adicional sobre las personas que participaron en la lectura de la tesis doctoral.

-Prioridad: Alta

-Derivados:

-Guardar información sobre tesis doctorales

-Identificador: RF-002-D1

-Tipo: Requisito funcional

-Descripción: Debemos dar la opción de guardar datos sobre tesis doctorales

-Prioridad: Alta

-Navegar hacia los doctores

-Identificador: RF-002-D2

-Tipo: Requisito funcional

-Descripción: Dada una tesis, se podrá navegar hacia las páginas personales de los doctores que participaron en la lectura de la tesis (Lector, Dirección, Tribunal)

-Prioridad: Alta

-Navegar hacia su institución

-Identificador: RF-002-D3

-Tipo: Requisito funcional

-Descripción: Deberemos poder navegar hacia las páginas de las instituciones en donde se hizo esa tesis doctoral

-Prioridad: Media

•**Visualizar en una pestaña información sobre instituciones**

-**Identificador:** RF-003

-**Tipo:** Requisito funcional

-**Descripción:** Deberá existir una página para mostrar información instituciones en las cuales se hayan presentado tesis doctorales y mostrar todas las tesis leídas allí.

-**Prioridad:** Media

-**Derivados:**

•**Guardar información sobre Instituciones**

-**Identificador:** RF-003-D1

-**Tipo:** Requisito funcional

-**Descripción:** Debemos dar la opción de guardar datos sobre

-**Prioridad:** Alta

•**Acceder a una pestaña de ayuda y contacto**

-**Identificador:** RF-004

-**Tipo:** Requisito funcional

-**Descripción:** Deberá existir una página en la que existan una serie de dudas frecuentes que pueden surgir a los usuarios junto con una explicación además de un formulario de contacto

-**Prioridad:** Alta

•**Acceder a una pestaña de inicio de sesión**

-**Identificador:** RF-005

-**Tipo:** Requisito funcional

-**Descripción:** Deberá existir una pestaña para poder iniciar sesión/registrarnos en el sistema.

-**Prioridad:** Alta

-**Derivados:**

•**Registro en el sistema**

-**Identificador:** RF-005-D1

-**Tipo:** Requisito funcional

- Descripción:** Deberemos proporcionar un sistema en el que los usuarios puedan crear cuentas de acceso
- Prioridad:** Alta

-Iniciar sesión utilizando ORCID

- Identificador:** RF-006
- Tipo:** Requisito funcional
- Descripción:** Deberá existir la posibilidad de utilizar tus datos en el sistema ORCID para poder registrarte o iniciar sesión en nuestro sistema
- Prioridad:** Media
- Derivados:**

-Modificar información

- Identificador:** RF-007
- Tipo:** Requisito funcional
- Descripción:** Deberá existir una forma de poder modificar la información sobre los doctores y tesis que tenemos en nuestro sistema
- Prioridad:** Alta
- Derivados:**

-Disponer de algún elemento que nos permita editar la información

- Identificador:** RF-007-D1
- Tipo:** Requisito funcional
- Descripción:** En las pestañas de visualización de tesis y doctores, los usuarios registrados podrán visualizar un botón para editar la información que le redirigirá a una interfaz de edición
- Prioridad:** Alta

-Visualizar una pestaña para edición de datos

- Identificador:** RF-007-D2
- Tipo:** Requisito funcional
- Descripción:** Deberemos incluir una pestaña en la que aparezca un formulario con los datos de tesis o doctores y la posibilidad de añadir modificaciones que serán aceptadas por el administrador

-Prioridad: Alta

-Acceder panel de administración

-Identificador: RF-008

-Tipo: Requisito funcional

-Descripción: Deberá existir una página en la cual solo pueda acceder el administrador para confirmar nuevos datos y modificaciones

-Prioridad: Alta

-Derivados:

-Confirmar modificación de tesis

-Identificador: RF-008-D1

-Tipo: Requisito funcional

-Descripción: Deberemos incluir un registro de las modificaciones de datos sobre tesis que podrá confirmar o no nuestro administrador

-Prioridad: Alta

- Confirmar modificación de doctores

-Identificador: RF-008-D2

-Tipo: Requisito funcional

-Descripción: Deberemos incluir un registro de las modificaciones de datos sobre doctores que podrá confirmar o no nuestro administrador

-Prioridad: Alta

-Confirmar nuevas tesis

-Identificador: RF-008-D3

-Tipo: Requisito funcional

-Descripción: Deberemos incluir un registro sobre nuevas inserciones de datos relativas a tesis que podrá confirmar o no nuestro administrador

-Prioridad: Alta

-Confirmar nuevos doctores

-Identificador: RF-008-D4

-Tipo: Requisito funcional

-Descripción: Deberemos incluir un registro sobre nuevas inserciones de datos relativas a doctores que podrá confirmar o no nuestro administrador

-Prioridad: Alta

-Eliminar tesis

-Identificador: RF-008-D5

-Tipo: Requisito funcional

-Descripción: Deberemos incluir un registro sobre peticiones de eliminación de datos relativas a tesis que podrá confirmar o no nuestro administrador

-Prioridad: Alta

-Eliminar doctores

-Identificador: RF-008-D6

-Tipo: Requisito funcional

-Descripción: Deberemos incluir un registro sobre peticiones de eliminación de datos relativas a doctores que podrá confirmar o no nuestro administrador

-Prioridad: Alta

5.2. Requisitos no funcionales

Los requisitos no funcionales no se corresponden con las funcionalidades explícitas que nos piden los clientes del proyecto, pero son imprescindibles porque se corresponden con funcionalidades que se deben incluir relacionadas con el uso de la aplicación, rendimiento, seguridad, ...

En este caso los propios requisitos no funcionales, los dividiremos en distintos grupos.

5.2.1. Requisitos de Aspecto

-Aspecto

-**Identificador:** RNF-001

-**Tipo:** Requisito no funcional

-**Descripción:** El sistema deberá proporcionar una interfaz intuitiva y clara. Se asegurará que el sistema cumple estos requisitos mediante los resultados de las pruebas realizadas mediante usuarios externos.

-**Prioridad:** Alta

-Simplicidad de uso

-**Identificador:** RNF-002

-**Tipo:** Requisito no funcional

-**Descripción:** Debemos intentar minimizar el número de interacciones por caso de uso

-**Prioridad:** Alta

5.2.2. Facilidad de uso

-Facilidad de uso

-**Identificador:** RNF-003

-**Tipo:** Requisito no funcional

-**Descripción:** El sistema deberá proporcionar información en el sistema sobre las acciones que puedes realizar dentro de este

-**Prioridad:** Alta

-Aspecto

-**Identificador:** RNF-005

-**Tipo:** Requisito no funcional

-**Descripción:** El sistema incluir un pequeño tutorial en el que se explique las funcionalidades disponibles y como realizarlas

-**Prioridad:** Alta

5.2.3. Requisitos operacionales

Seguridad

-Identificador: RNF-006

-Tipo: Requisito no funcional

-Descripción: El sistema deberá encriptar la información guardada de modo que ante un acceso no autorizado al servidor no puedan leer obtener información privada.

-Prioridad: Alta

·Velocidad

-Identificador: RNF-007

-Tipo: Requisito no funcional

-Descripción: El sistema deberá poder utilizarse de forma fluida de modo que la experiencia del usuario no se vea afectada. El retraso máximo aceptado será de 2 segundo debido a las especificaciones del alojamiento

-Prioridad: Alta

·Disponibilidad

-Identificador: RNF-008

-Tipo: Requisito no funcional

-Descripción: El sistema deberá estar disponible para su acceso en todo momento

-Prioridad: Alta

5.2.4. Requisitos de mantenimiento y portabilidad

·Portabilidad

-Identificador: RNF-011

-Tipo: Requisito no funcional

-Descripción: El sistema deberá ser accesible desde cualquier dispositivo con acceso a internet y la tecnología de alojamiento ni de desarrollo deben restringir este acceso

-Prioridad: Alta

5.2.5. Requisitos legales

-Uso de datos externos

-**Identificador:** RNF-012

-**Tipo:** Requisito no funcional

-**Descripción:** El sistema hará uso de información suministrada por la API de ORCID siempre que los usuarios autoricen este intercambio.

-**Prioridad:** Alta

-Uso de cookies de sesión

-**Identificador:** RNF-013

-**Tipo:** Requisito no funcional

-**Descripción:** El sistema utilizará la tecnología de Cookies de sesión para mejorar la experiencia del usuario. Este tipo de cookies no se ven afectadas por los cambios en la legislación reciente.

-**Prioridad:** Alta

5.3. Requisitos de documentación

Este apartado incluirá la documentación adicional necesaria para finalizar el proyecto. Estos documentos suelen variar según el tipo de proyecto ya que, por ejemplo, en nuestro caso en el que trata de una aplicación web no necesitará un manual de instalación o de requisitos de compatibilidad, pero sí un documento que guíe a los usuarios sobre el funcionamiento de nuestro sistema.

Manual de usuario:

El manual de usuario de nuestro sistema vendrá incluido en nuestra propia aplicación web. De una manera breve y sencilla el usuario encontrará en la página principal un pequeño texto introductorio junto con una guía sobre las funcionalidades del sistema.

Este manual introductorio, lejos de ser un índice sobre la estructura del sistema,

funcionará como una guía de uso que tendrá el apoyo de más documentación en caso de dudas o incidencias.

Ayuda y contacto:

Pestaña adicional del sistema que contará con las dudas más frecuentes que pueden surgir durante el acceso a nuestra página web y con la incorporación de un formulario de ayuda para que un administrados o encargado, trate de solucionar dudas o incidencias de los usuarios.

Este formulario, llevará el soporte de un correo electrónico asociado al sistema desde el cual, el administrador podrá responder dudas o incidencias sin tener que utilizar ningún medio personal.





6

Análisis y gestión de riesgos

Un riesgo es un evento que puede suceder en cualquier momento de la ejecución de nuestro sistema y pueden tener un impacto negativo en cualquier aspecto de nuestro sistema.

Es por ello, que identificar todos los posibles riesgos que puedan producirse y planificar una estrategia de mitigación de daños en caso de que se produzcan puede servir para favorecer el mantenimiento de nuestro sistema.

En un proyecto software podemos distinguir entre dos tipos de riesgos:

-Riesgos del proyecto: Son riesgos que afectarán al desarrollo del proyecto y que están relacionados con la planificación, la gestión de recursos o la integridad del proyecto.

-Riesgos técnicos: Son riesgos relacionados con la funcionalidad del producto resultante debido a problemas con la definición de requisitos, elección de las tecnologías o movimientos de la competencia.

La estrategia a seguir ante cada riesgo vendrá predeterminada por dos factores:

-Probabilidad: la probabilidad es la posibilidad de que un evento se produzca durante el desarrollo del proyecto o bien durante su funcionamiento. Usaremos una escala para valorar su probabilidad que será Baja, Media, Alta.

-Impacto: El impacto valorará el efecto negativo que producirá el evento. Para asignar un impacto utilizaremos la escala Bajo, Medio, Alto.

Ahora se presentarán los riesgos del proyecto en la tabla 2 y los riesgos técnicos en la tabla 3

Requisitos	Impacto	Probabilidad	Descripción
Fallo en la planificación	Medio	Media	Un fallo en la planificación deberá solucionarse con un intento de trabajar más horas con el fin de cumplir la fecha de entrega.
El personal abandona la empresa	Medio	Media	Se deberá buscar otro trabajador de parecidas cualidades y ofrecerle unas buenas condiciones para que no abandone el proyecto
El personal no puede llevar a cabo las funcionalidades pedidas	Medio	Baja	Se estudiará entre si es más factible aumentar el tiempo de planificación, proporcionar cursos de instrucción a los empleados o contratar nuevos empleados.
Faltan recursos	Medio	Baja	Se adquirirán más recursos sin mayor dilación para garantizar el cumplimiento del contrato

*Tabla 2: Riesgos del proyecto

Requisitos	Impacto	Probabilidad	Descripción
Objetivo no se puede cumplir por fallo en la definición de requisitos	Alto	media	Como se utiliza una metodología ágil se deberán rediseñar los requisitos de modo que se acabe cumpliendo el objetivo
Fallo en la implementación	Bajo	Media	Volver a programar esa parte del software con especial atención en que cumpla las pruebas de forma correcta
El servidor queda inactivo temporalmente	Alto	Baja	Si se prevé que puede poner en peligro nuestro proyecto se deberá buscar otro sitio para alojar nuestro sistema
Golang deja de tener soporte	Medio	Baja	Para futuras modificaciones debería hacerse una migración hacia otra tecnología
El PaaS usado deja de funcionar	Medio	Baja	Se deberá alojar el sistema en otro sistema que permita el uso de las tecnologías utilizadas
El administrador se va y no desea continuar a cargo del proyecto	Medio	Media	Se deberá asignar a otro administrador y se le deberá dar acceso total al PaaS donde está instalado el sistema y a la base de datos

Aparece otra página web con un funcionamiento idéntico	Alto	Media	En caso de que el flujo de actividad en nuestro sistema sea muy bajo se cerrará la aplicación
Nadie desea hacerse cargo del proyecto web	Alto	Media	Se cerrará la aplicación por si en algún futuro alguien desea utilizarla

*Tabla 3: Riesgos técnicos

Modelado y diseño

La fase de modelado y diseño del software es la segunda fase del desarrollo software y es la fase en la que, disponiendo ya del análisis de los objetivos y especificaciones del sistema, se realizarán una serie de abstracciones que servirán de guía en el desarrollo del software del sistema.

Estas abstracciones, no solo consisten en el diseño de la arquitectura software en la que se basará el sistema ya que también tratará de modelar otros aspectos del sistema como los casos de uso, un modelo de navegación de la aplicación web o un modelo de componentes en los que se definirán la situación física de los componentes.

Para el desarrollo de esta fase, se ha utilizado la herramienta MagicDraw, <https://www.magicdraw.com>, la cual, disponemos de una licencia a través de la universidad de Málaga que nos permite su uso para realizar trabajos con fines académicos.

Con esta herramienta, hemos necesitado una extensión llamada UML-Base Web Engineering (UWE), que nos permite definir instancias de elementos web en nuestro modelo para su mejor comprensión.

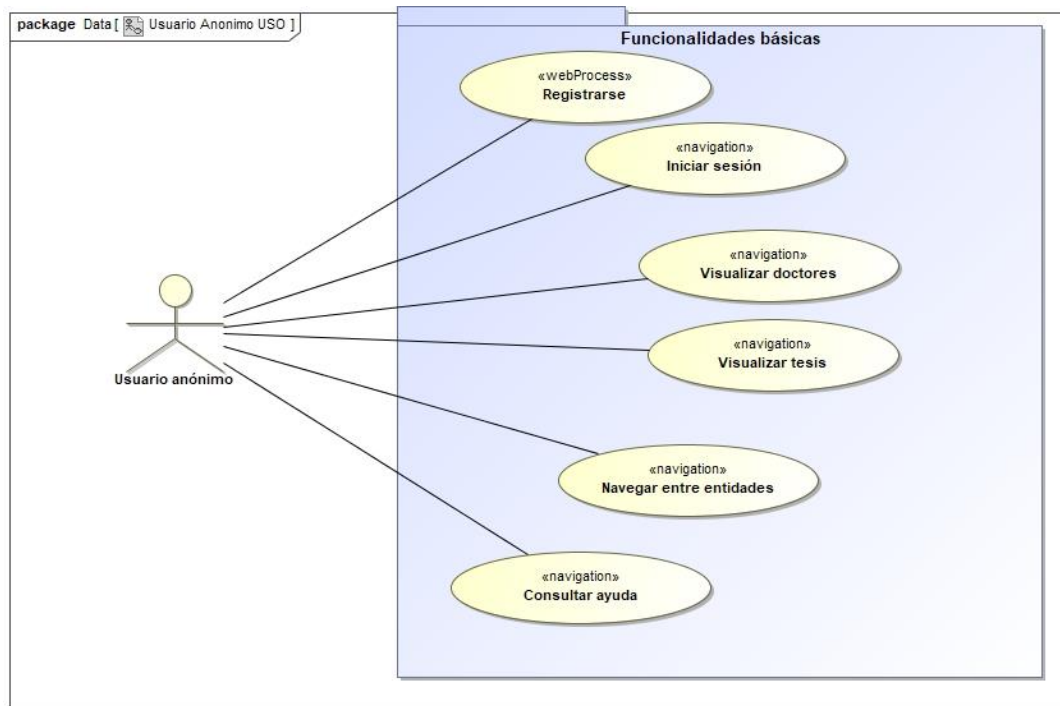
7.1. Casos de uso

Los diagramas de caso de uso son los idóneos para comenzar con el modelado de nuestro sistema ya que estos diagramas nos permiten describir los posibles escenarios para los que un sistema se ha desarrollado. Estos diagramas representan lo que el sistema debería realizar sin incluir detalles de la implementación, algoritmos, etc. (Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel, UML@Classroom, 2012)

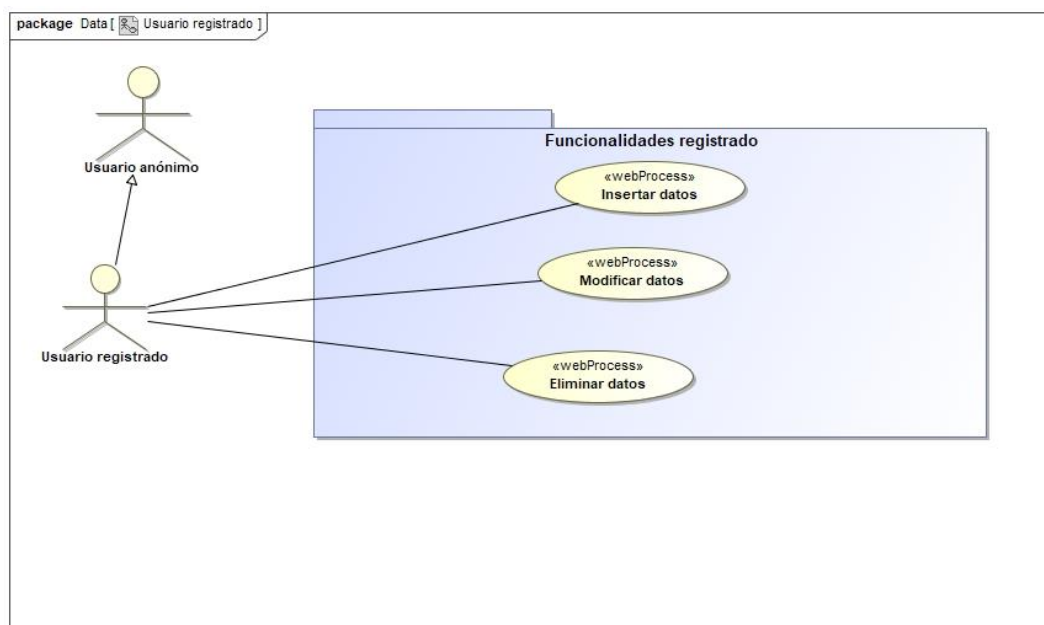
En nuestro caso, disponemos de más de un rol de usuario con funcionalidades distintas cada uno por lo que para su mayor comprensibilidad, generaremos un diagrama de uso por cada diferente tipo de usuario.

Además, usaremos varios conceptos específicos en el modelado enfocado a ingeniería web como son el uso de los estereotipos <<**webProcess**>> que significa que es un proceso que se realiza en el sistema pero que no conlleva ninguna redirección y <<**navigation**>> que significa que el usuario está realizando alguna redirección dentro del sistema.

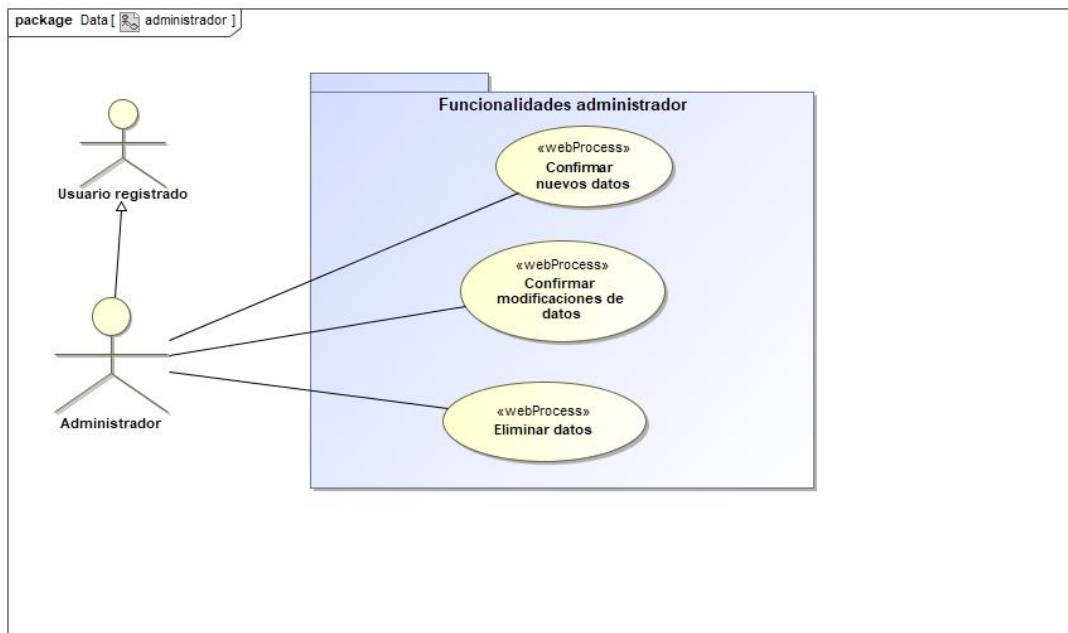
A continuación, se muestra el caso de uso perteneciente a los usuarios anónimos, es decir, los usuarios que acceden sin identificarse en nuestra aplicación web, diagrama 1. El diagrama de casos de uso perteneciente a los usuarios identificados en el sistema, diagrama 2. Finalmente, el diagrama de casos de uso asociado a los administradores en el diagrama 3.



*Diagrama 1: Diagrama de casos de uso de usuarios anónimos.



*Diagrama 2: Diagrama de casos de uso de usuarios identificados.



*Diagrama 3: Diagrama de casos de uso de administradores

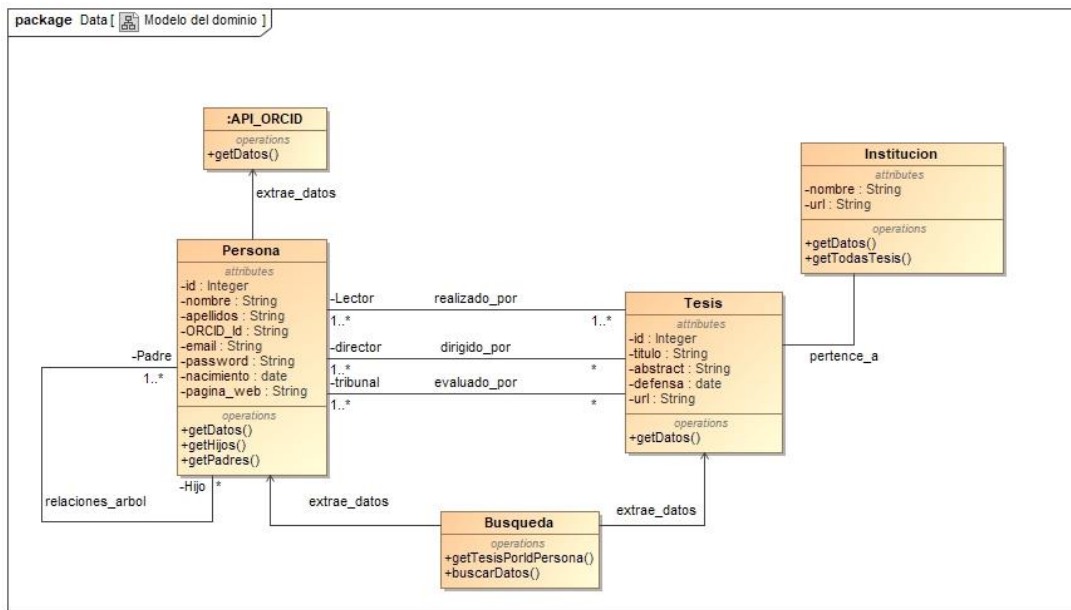
Como podemos ver, los casos de usos están recogidos en paquete de modo que el usuario registrado aparte de disponer del paquete de funcionalidades registradas también tiene relación con las funcionalidades básicas y a su vez, el administrador tiene relación con todos los paquetes.

7.2. Diagrama de clases

Usamos los diagramas de clases para modelar la estructura estática de un sistema. Pero, además de describir los elementos del sistema, el diagrama de clases muestra las relaciones entre estos elementos.

Se presupone, que estos elementos y sus relaciones no variarán respecto al tiempo. (Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel, UML@Classroom, 2012)

A continuación, se muestra el diagrama de clases perteneciente a nuestro sistema



*Diagrama 4: Diagrama de clases.

Como entidad controladora, tenemos la entidad “Búsqueda” que es la que se encarga de coger información de tesis doctorales y doctores para los usos de la aplicación.

El fuerte acoplamiento entre la entidad “Persona” y la entidad “Tesis” viene precedido porque ambas instancias representan información del otro cuando son visualizadas, esto provoca que en todos momentos tengan que estar en contacto mutuo.

Además, las relaciones entre tesis y su institución sirven a modo de guardar las relaciones entre las personas que participaron en la lectura de una tesis doctoral especificando el rol de cada persona.

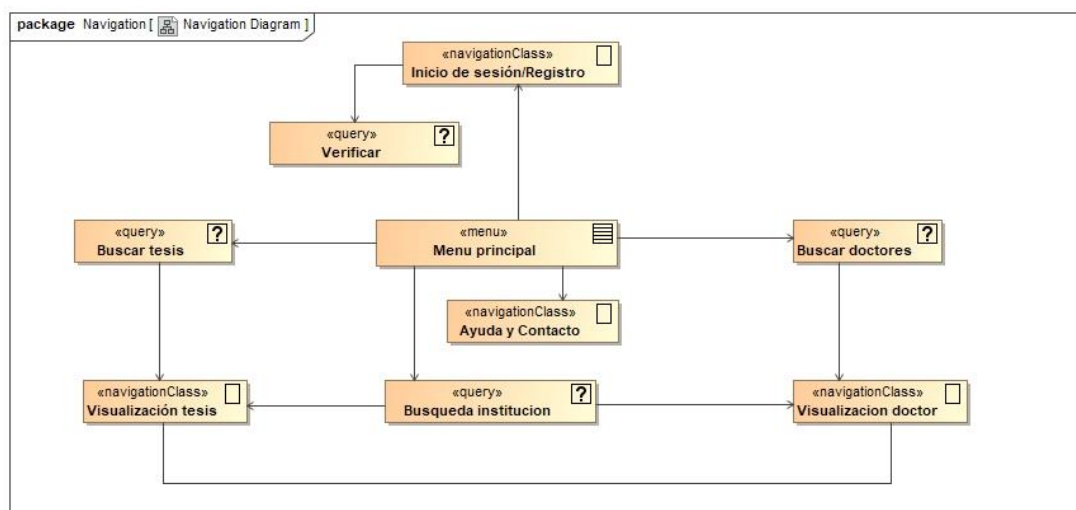
Por su parte la entidad Persona contiene unas relaciones propias para poder generar su árbol genealógico sin necesidad de recurrir a las relaciones entre tesis

Finalmente, la entidad API_ORCID representa el acceso a la API con el fin de coger información de doctores desde su sistema para su registro en nuestro propio sistema. Esto lejos de ser algo constante, se realiza solo una vez por usuario y solo si el usuario accede mediante su cuenta ORCID.

7.3. Modelo de navegación

Ya que hemos podido visualizar como se estructura el sistema internamente en el lado del servidor, el modelo de navegación se encargará de representar el funcionamiento de la interfaz del sistema. Este diagrama no tratará de mostrar simplemente las vistas, sino que representará las navegaciones que se podrán efectuar sobre el sistema y los tipos de elementos que contendrá el sistema.

A continuación, presentamos el diagrama de navegación de nuestro proyecto:



*Diagrama 5: Modelo de Navegación

Antes de comenzar con la explicación del modelo de navegación perteneciente a este proyecto definiremos algunos indicadores del diagrama para ayudar a su comprensibilidad:

<<Query>>: Trata de un aspecto de la interfaz cuya única funcionalidad es la de hacer una consulta a la base de datos para mostrar otra interfaz.

<<Navigation class>>: Trata de una interfaz en la que principalmente

se muestra información. Esto no significa que este elemento no pueda contener redirecciones hacia otras páginas, pero muestra que su principal objetivo es el de mostrar información.

<<Menu>>: Trata de un conjunto de enlaces hacia otras interfaces de mayor importancia. Este elemento sirve de guía para la redirección y navegabilidad dentro de nuestro sistema y así poder acceder a otras vistas en todo momento.

Una vez definidos estos indicadores, procedemos a explicar algunos puntos del modelo.

En este modelo como punto principal se va a remarcar un detalle que resulta primordial para comprender el funcionamiento de nuestro sistema. Al contrario que en muchas páginas web, en nuestro sistema se nos permite acceder al contenido sin tener que registrarnos. Es por ello, que el menú principal redirige hacia la pestaña de Inicio de sesión/Registro y no al revés.

Seguido de ello, podemos comprobar que desde el menú principal podemos acceder a distintas interfaces así que vamos a proceder a explicar todas:

-Búsqueda tesis/doctor: Podemos encontrar una simetría en estos fragmentos del modelo y es que ambos comparten el mismo funcionamiento en el que mediante una búsqueda podemos seleccionar un doctor o una tesis y a su vez, navegar hacia la tesis que realizó ese doctor o hacia los doctores que participaron activamente en la realización de la tesis.

-Búsqueda institución: Aunque también comparte funcionamiento con la búsqueda de doctores o tesis, esta búsqueda nos redireccionará a una lista de tesis y doctores entre los cuales podremos seleccionar cual queremos visualizar.

-Ayuda y contacto: Vista que mostrará dudas frecuentes de los usuarios y un formulario de contacto con el administrador del sistema para resolución de incidencias.

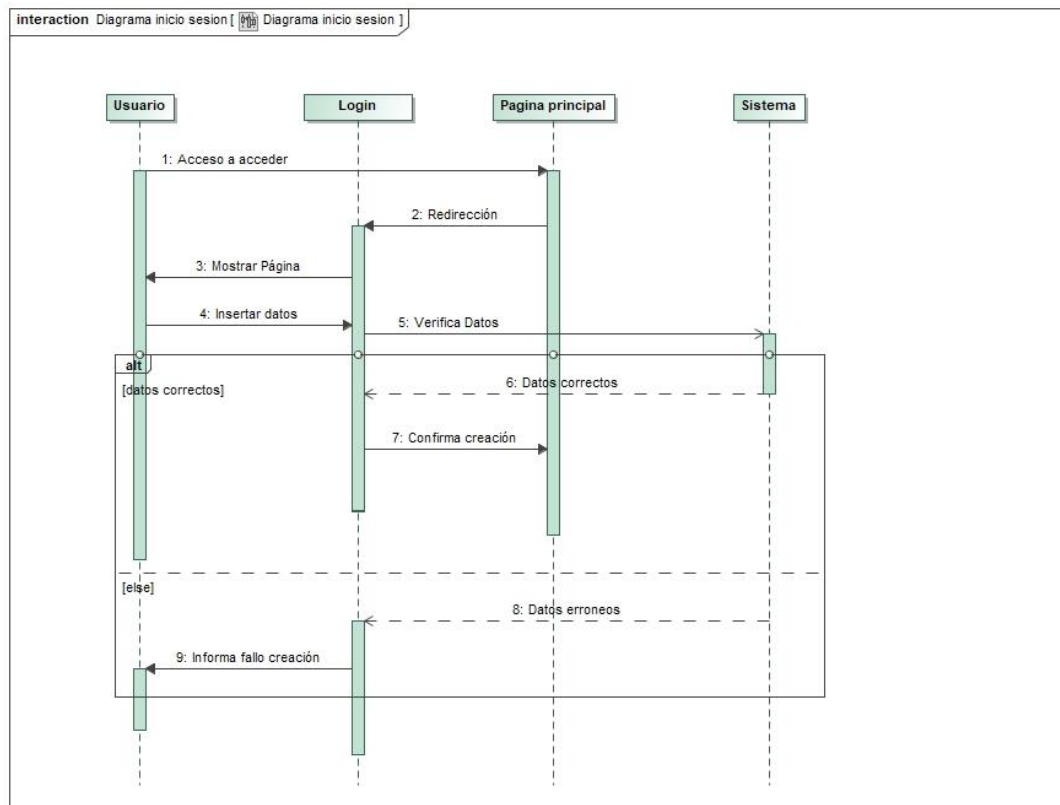
7.4. Diagramas de secuencia

Ya hemos visto los modelos que se corresponden con el funcionamiento del sistema desde un aspecto abstracto a través de las entidades que utilizará, del servidor o de la propia interfaz así que ahora corresponde explicar el comportamiento del sistema en donde se mostrará las interacciones de los usuarios con el sistema para llevar a cabo cada caso de uso.

A continuación, se procederá a mostrar los diagramas de secuencia pertinentes al proyecto actual y su explicación dividido por los roles de usuario que podrán hacer uso de ellos (Comprendemos que conforme más alto es el rol, las funcionalidades su inclusivas):

Usuario anónimo:

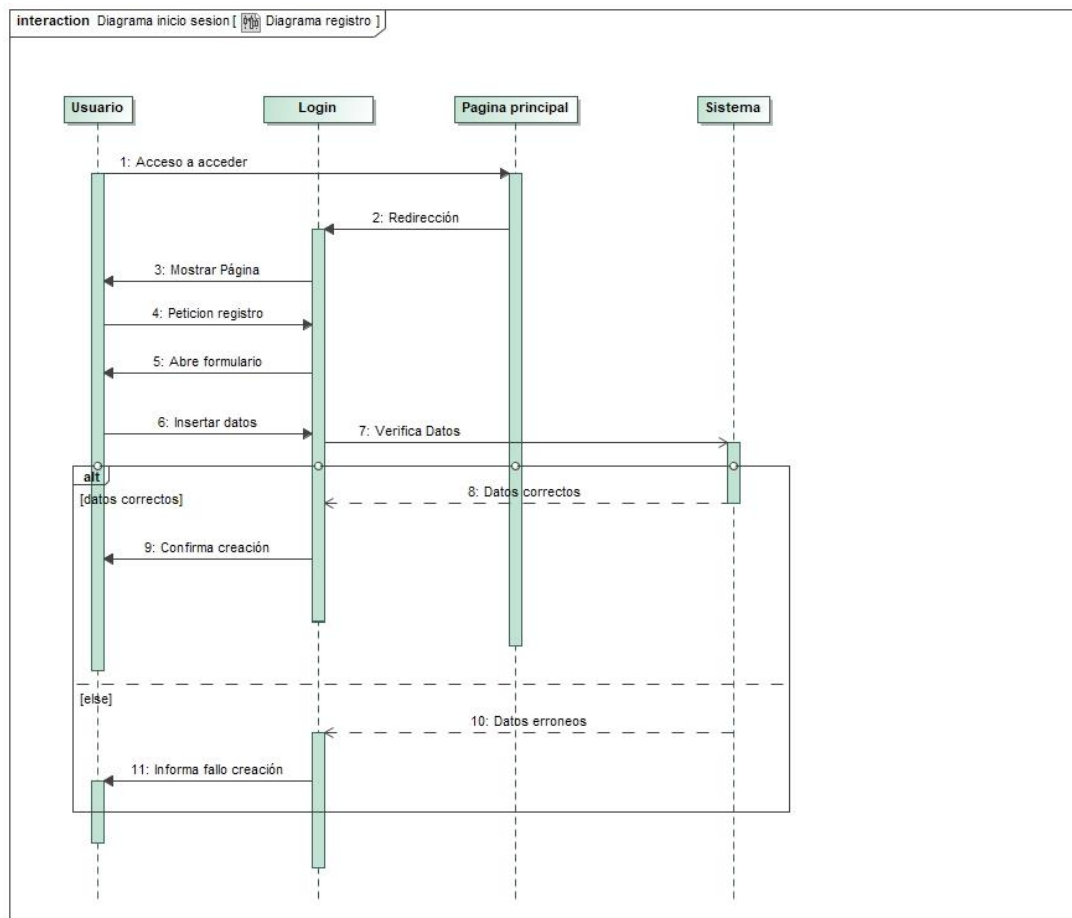
·Inicio de sesión:



*Diagrama 6: Diagrama de secuencia de inicio de sesión

Ejemplo realizado desde la página principal, pero al tener un menú permanente en la interfaz se podrá realizar desde cualquier ventana

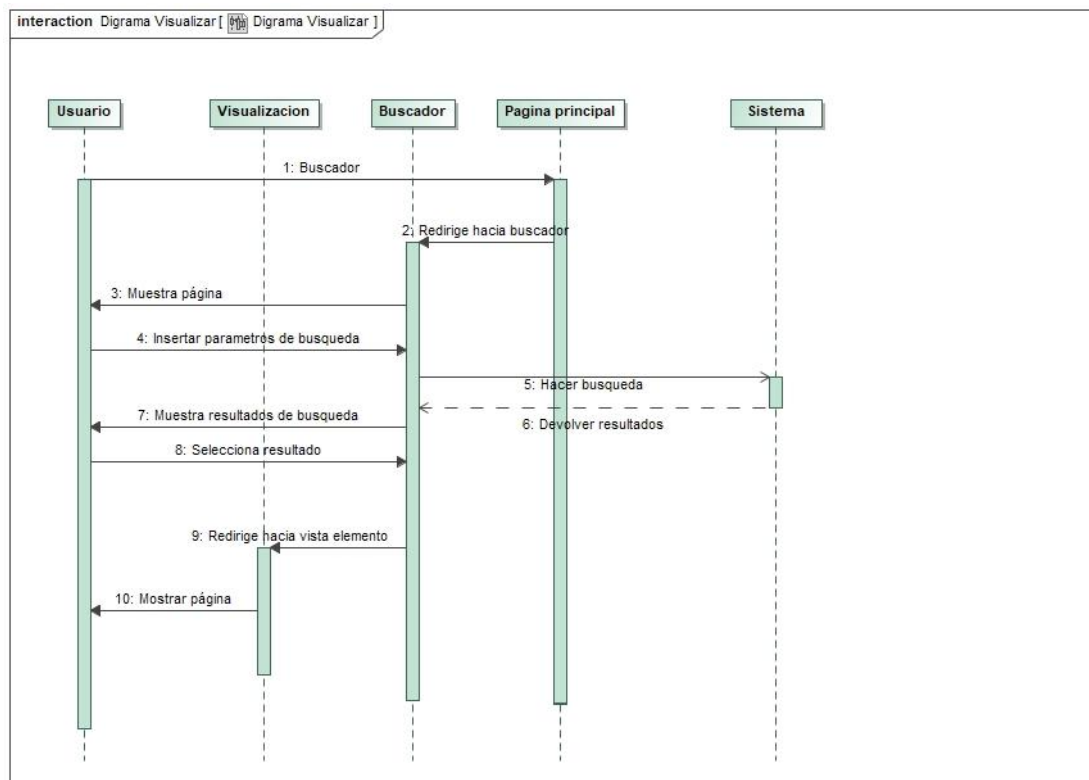
•Registro



*Diagrama 7: Diagrama de secuencia de registro

Ejemplo realizado desde la página principal, pero al tener un menú permanente en la interfaz se podrá realizar desde cualquier ventana

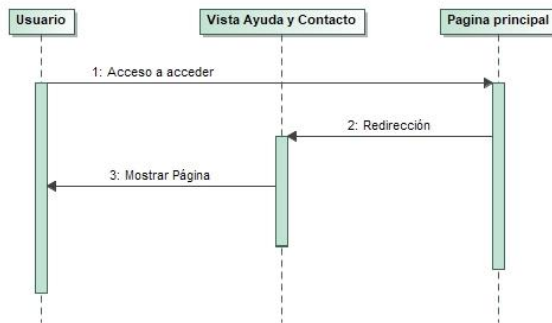
•Visualizar dato



*Diagrama 8: Diagrama de secuencia visualizar dato

Vemos como los usuarios deberán realizar una búsqueda en nuestro sistema antes de poder acceder a un sistema. El acceso a la herramienta de búsqueda no está restringido por lo que los usuarios no identificados podrán realizar esta funcionalidad sin ningún problema.

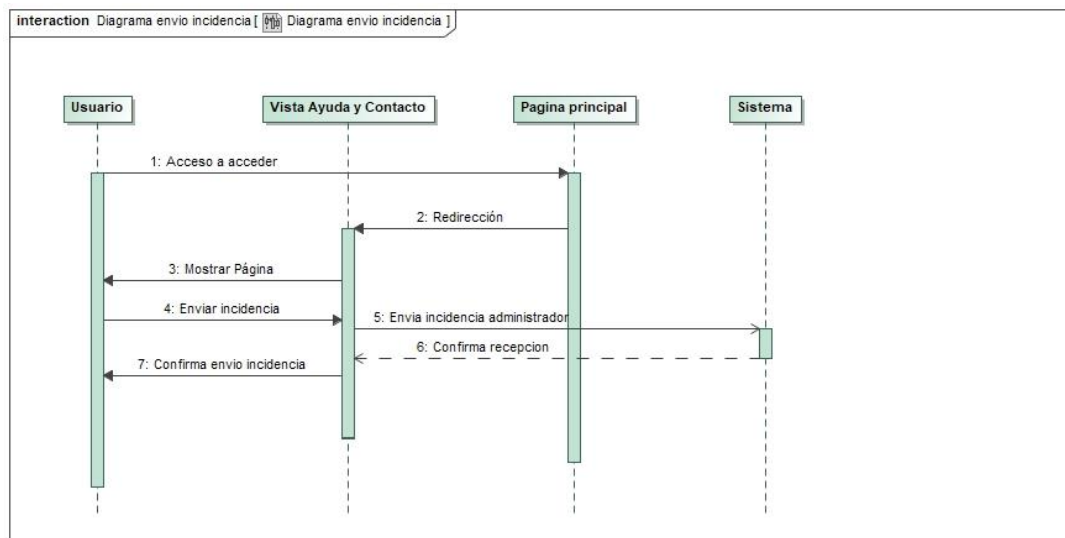
•Buscar ayuda:



*Diagrama 9: Diagrama de secuencia de buscar ayuda

Ejemplo realizado desde la página principal, pero al tener un menú permanente en la interfaz se podrá realizar desde cualquier ventana

•Enviar incidencia

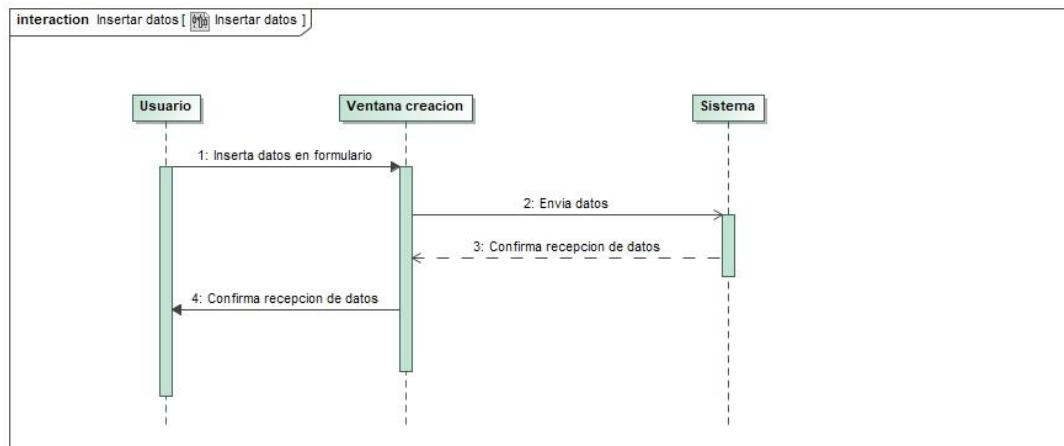


*Diagrama 10: Diagrama de secuencia de envío de incidencia

Ejemplo realizado desde la página principal, pero al tener un menú permanente en la interfaz se podrá realizar desde cualquier ventana. No hará falta registrarse en el sistema, pero necesitará poner un email de contacto en el que enviar la respuesta.

Usuario registrado:

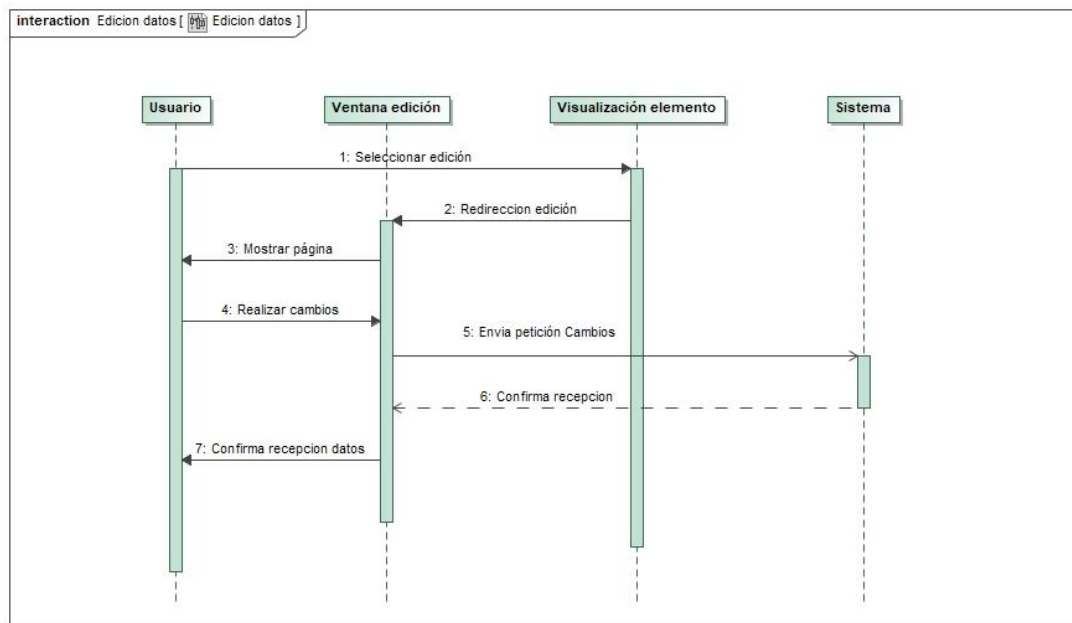
-Inserción de datos



*Diagrama 11: Diagrama de secuencia de inserción de datos

Comenzando en la página de inserción de datos, el usuario deberá rellenar los campos de inserción pertinentes y enviar los datos, una vez que estos se insertan, recibirá un mensaje de confirmación el usuario. Estos datos deberán ser confirmados por un administrador para que sean aceptados.

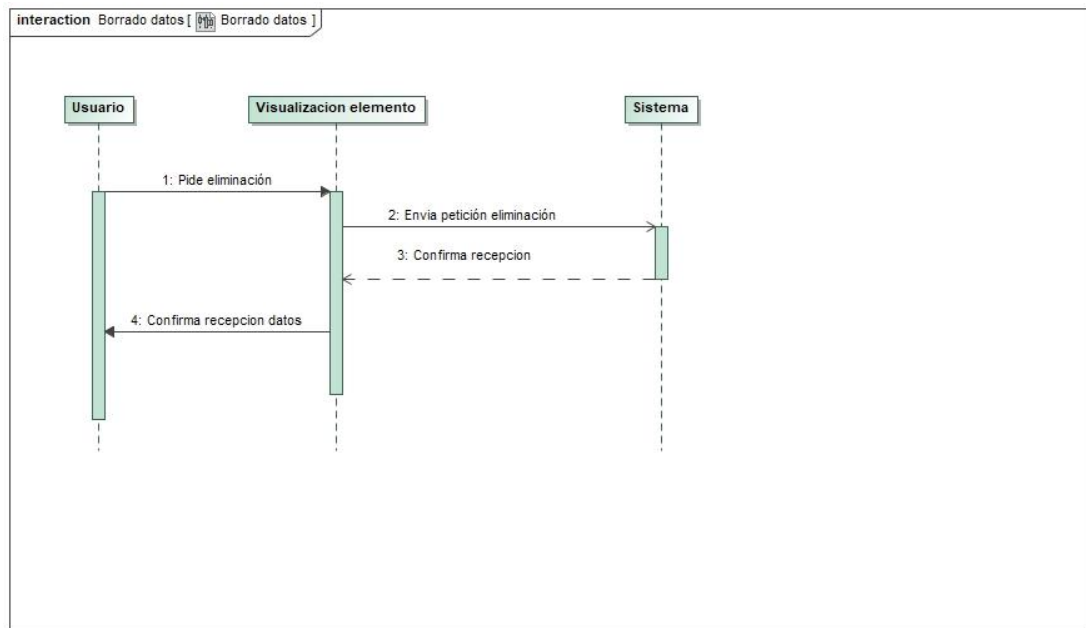
•Edición de datos



*Diagrama 12: Diagrama de secuencia de edición de datos

El usuario accederá a la ventana de inserción de datos y a través de esta podrá insertar las modificaciones que crea pertinentes. Estas modificaciones serán enviadas y confirmadas por un administrador antes de verse modificadas en el sistema.

•Eliminar datos

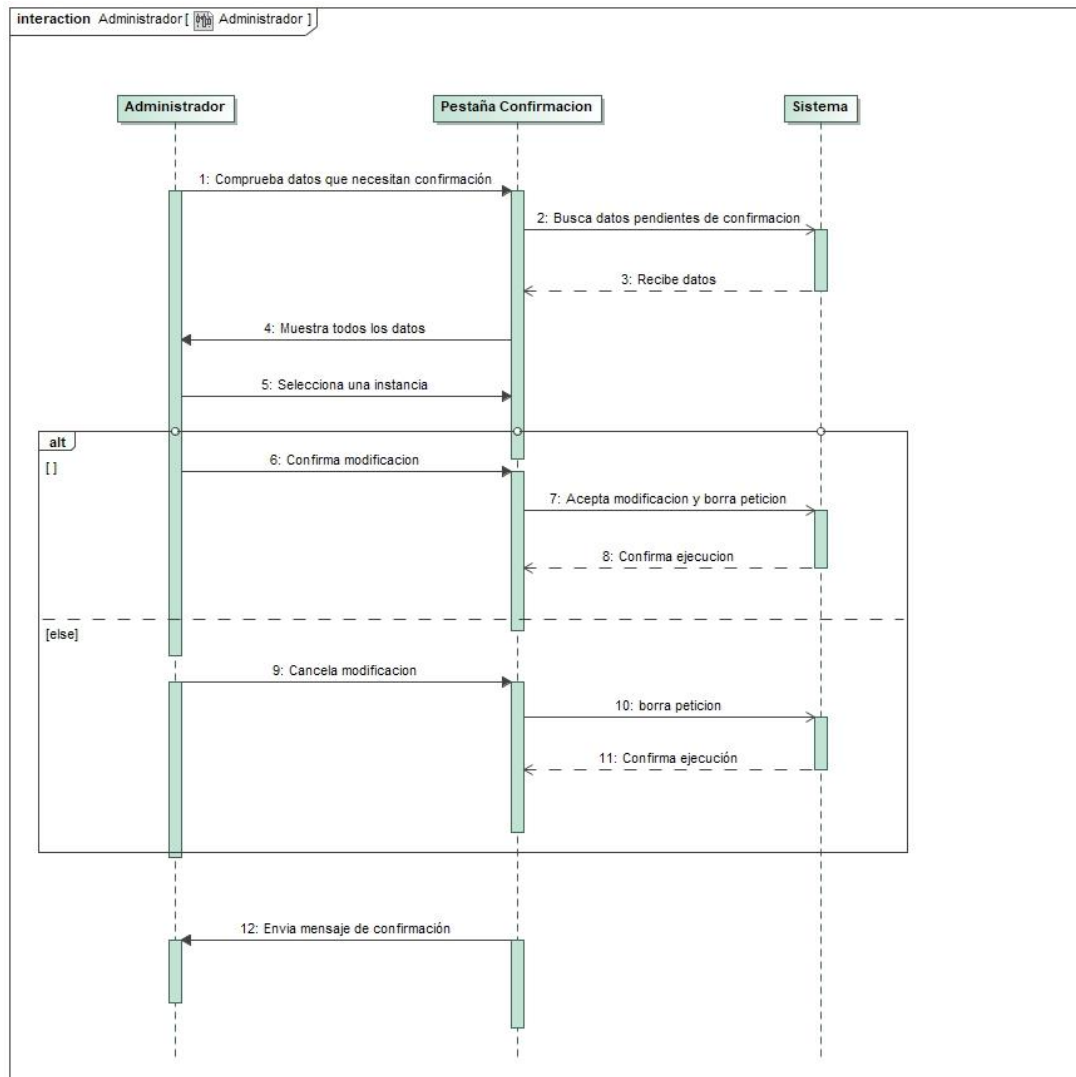


*Diagrama 13: Diagrama de secuencia sobre eliminar datos

El usuario a través de la pestaña de visualización de la información dispondrá de un botón para pedir que se elimine este dato. El administrador tendrá que confirmar la eliminación definitiva del dato.

Administrador:

-Confirmación/edición/eliminación datos



*Diagrama 14: Diagrama de secuencia de panel de administrador

El administrador realizará el mismo procedimiento para aceptar nuevos datos, modificar datos y eliminar datos definitivamente.



8

Implementación

Entrando en la tercera fase del desarrollo del proyecto, pasamos a la implementación del software. En esta fase, los analistas y programadores deberán ceñirse a la documentación pertinente y seguir la estructura diseñada por el arquitecto software. De este modo, tras la ya elección de las tecnologías a usar, desarrollarán el código de un modo planificado en el que tendrán constancia de las dependencias del propio código.

8.1. Estructura general

Para el desarrollo de este proyecto, debido a la inexperiencia con la tecnología utilizada, pero habiendo utilizado tecnologías similares, el autor decidió simular la estructura que realizó en varios proyectos desarrollados con Node JS.

Esta estructura consiste en que, para organizar el servidor, se utilizarían 3 ficheros esenciales aparte del fichero donde empieza a correr el sistema: Router, Handler, Database.

-Router: Contiene el “listener” y tratará de redirigir cada acceso a una URL diferente con el método correspondiente. Estos accesos a direcciones URL, se presentarán de distintos modos en nuestro sistema como peticiones GET y POST.

-Handler: Contiene todos los métodos de los que hará uso la clase “Router” y por lo tanto es la que se encargará tanto de cargar vistas al acceder a un sitio, como de llamar a métodos que hagan llamadas a la base de datos.

-Database: Utilizado para tener un control sobre la base de datos y que no se diferentes conexiones que queden abiertas indefinidamente, esta clase utilizará un patrón “Singleton” y contendrá todos los métodos utilizados para hacer llamadas a la base de datos.

En el lado del cliente, la estructura es simple, organizamos todo el código en tres carpetas diferentes: Javascripts, Diseños, Vistas

-Javascripts: Carpeta donde se encuentran todos los ficheros con extensión .js que se utilizarán en las vistas

-Diseños: Carpeta donde se encuentran todos los ficheros con extensión .css que se utilizarán en las vistas

-Vistas: Carpeta donde se encuentras los ficheros con extensión .html

8.2. Tecnologías fundamentales

Un aspecto que puede llegar a ser muy determinante en cualquier proyecto software es la correcta elección de las metodologías. En este caso, la tecnología base que es la utilizada para el desarrollo del servidor, fue prefijada como requisito, por lo tanto, la elección de otras herramientas ha venido delimitadas por Golang.

A continuación, se hará un breve resumen de las tecnologías básicas utilizadas y de las ventajas y desventajas que nos presenta:

-**Golang** o **Go**, es un lenguaje de programación desarrollado por Google en 2009. Este lenguaje, a pesar de ser prácticamente nuevo ha ido ganando popularidad últimamente debido a algunas características que lo diferencian de la competencia como la posibilidad de utilizar sistemas concurrentes, utilizar punteros o funciones que devuelven más de un parámetro.

-**PostgreSQL** es un sistema gestor de base de datos (SGBD) relacional orientado a objetos y de código abierto. La elección de este SGBD se debe a la posibilidad de disponer de un sistema de almacenamiento de objetos que disponga de un rendimiento aceptable en función de la velocidad y capacidad y que sea gratuito.

-**Javascript** es uno de los lenguajes más usados en el desarrollo del front-end de aplicaciones web. Muchas veces, nos lo encontramos con extensiones como TypeScript o Angular Js pero esta vez no se han utilizado para no incrementar la complejidad del desarrollo debido al desconocimiento de estas tecnologías.

-**GIT** tecnología imprescindible actualmente utilizada para el control y mantenimiento de versiones en la cual a través de un repositorio común he podido ir guardando y controlando las distintas versiones del proyecto

8.3. Tecnologías adicionales

Normalmente, en los proyectos software, se utilizan un conjunto de librerías y frameworks que nos ayudan a mantener el código limpio y comprensible permitiéndonos reutilizar procedimientos ya existentes que mejoran incluso la funcionalidad de algunos elementos de nuestra interfaz.

Aun así, el uso de librerías y frameworks de terceros puede convertirse en un problema ya que, al incluir su código en el nuestro, estamos incluyendo posibles vulnerabilidades o malas prácticas de programación que puede afectar también al rendimiento de nuestro producto por lo que no son muy aconsejables para

ciertos tipos de proyectos.

Por esta razón, se recomienda activamente en caso de utilizar librerías externas, utilizar versiones estables testadas contra vulnerabilidades y de fuentes fiables que nos aseguren que ese código no ha sido manipulado

En este proyecto el conjunto de librerías y frameworks externos usados son:

- Bootstrap 4**, utilizado para facilitar el diseño web de nuestra aplicación ya que cuenta con plantillas sobre objetos que mejora el aspecto de elementos básicos.

- JQuery**, una librería Javascript de las más utilizadas que nos permite facilitar la programación incluyendo en sí la tecnología Ajax que nos permite realizar llamadas asíncronas.

- Vis.js**, un framework que se ha utilizado para la representación de grafos en nuestra interfaz gráfica

- Datatable js**, un framework utilizado para la creación y manipulación de tablas dinámicas en nuestra página de modo que no importe el desconocimiento de la cantidad de elementos que se mostrarán finalmente.

8.4. Despliegue

Una vez se finaliza con el desarrollo del software, se procede al despliegue de la aplicación. En el proceso de despliegue podemos encontrar varios puntos que pueden limitar nuestras opciones como pueden ser la compatibilidad de versiones y tecnologías o el coste del alojamiento.

Como opción ante el problema de compatibilidad de versiones recientemente ha ganado popularidad la tecnología Docker la cual crea contenedores donde

incluye todas las dependencias necesarias para ejecutar el programa y así encontrar menos dificultades a la hora de alojar su sistema software.

Otra opción viable a la hora de priorizar la compatibilidad de versiones y tecnologías es la de contratar una máquina virtual online desde la cual replicar tu entorno de desarrollo y ejecutar ahí tu proyecto de modo que se pueda acceder desde cualquier sitio.

En este proyecto, se ha decidido utilizar una máquina virtual alojada en la nube, lo que se conoce como Infraestructura como servicio (IaaS). En este caso replicamos un sistema con Ubuntu 16.04 LTS, Golang 1.10.1 y Postgres 1.15. A partir de ahí se nos proporciona una IP de acceso a través de la cual podremos acceder remotamente mediante ssh para acceder al terminal de la máquina virtual o a través de la URL junto el puerto activo para acceder a la aplicación web que se ejecuta en ese sistema.

El IaaS utilizado en esta ocasión se trata de <https://www.digitalocean.com/>, la cual mediante una versión de prueba hemos podido alojar nuestra aplicación web en la dirección (<http://46.101.0.69:8000/>). De este modo, conseguimos que el sistema este operativo y accesible en todo momento.



9

Mantenimiento y pruebas

Finalmente, la última parte perteneciente a nuestra metodología ágil iterativa se corresponde con el análisis de pruebas y mantenimiento.

Las actividades relacionadas con el aseguramiento de la calidad juegan un papel importante en la industria de software debido a su participación a través de todo el ciclo productivo. El control de la calidad de los productos o servicios desde etapas tempranas de los proyectos permite minimizar los costos asociados con esfuerzo y tiempo, lo que implica para el negocio un aumento de la calidad y satisfacción de los clientes. (Jaime Echeverri, Miguel Aristizábal, Liliana González, 2013)

Es por esto, que resulta imprescindible llevar a cabo un conjunto de pruebas para detectar posibles fallos tanto en la implementación como en el diseño de la interfaz lo que nos permite llevar a cabo modificaciones para mejorar la calidad del producto.

9.1. Consideraciones previas

A la hora de llevar a cabo las pruebas conviene hacer un análisis para determinar qué tipo de pruebas serían las más convenientes a realizar en nuestro proyecto.

Desde este punto de vista, el autor cree conveniente superponer que los usuarios realicen pruebas sobre la aplicación web para evaluar la calidad de la interfaz ya que al valorar el sistema personas ajenas al desarrollo, utilizarán una perspectiva más objetiva y se comprobará la calidad de la interfaz de usuario y la facilidad de uso del sistema.

9.2. Test Experimental

Para la realización de las pruebas se ha generado un test el cual constará de 20 preguntas. Estas 20 preguntas estarán divididas en 3 grupos:

-Interfaz: valorará la calidad de la interfaz respecto a la ubicación dentro del sistema y el diseño

-Funcionalidades: Comprobará si comprenden las funcionalidades que puede realizar y cómo

-Facilidad de uso: Valorará los procedimientos que tiene que realizar para llevar a cabo una funcionalidad.

Estas preguntas estarán redactadas de una manera simple e informal para poder realizar el test con personas no cercanas a la terminología informática.

Cada pregunta tendrá una serie de posibles opciones de respuesta, en las que cada pregunta tendrá un valor en función de la opción escogida en la respuesta. Además, al final del test tendrán la posibilidad de insertar una pequeña reseña sobre sus impresiones.

Para calcular el resultado del test, se sumarán todos los valores de cada

pregunta por grupos y se valorará la calidad de cada aspecto de nuestra aplicación web.

A continuación, en las figuras 1, 2 y 3 puede encontrar el test realizado por los usuarios.

Interfaz

¿Cree que es fácil situarse en página se encuentra dentro de la aplicación?

- ☐ 0. Sé en todo momento en donde me encuentro
- ☐ 1. A veces, no sé dónde estoy, pero sé cómo volver
- ☐ 2. Me pierdo a menudo, no sé localizarme bien

¿Cree que es una interfaz muy recargada?

- ☐ 0. Para nada, la pantalla está muy clara
- ☐ 1. La cantidad de elementos en la pantalla es la justa
- ☐ 2. Estaría bien que hubiera un poco menos de contenido

¿Cree que la interfaz es demasiado básica?

- ☐ 0. Muestra lo justo y necesario. Está bien
- ☐ 1. Se podría mejorar un poco el diseño
- ☐ 2. Se ve demasiado vacía la pantalla

*Figura 1: Apartado Interfaz

Aquí vemos 3 cuestiones en las que cada una se evalúa con un número entre 0 y 2 de modo que se buscará conseguir la menor valoración posible en los resultados del test

Funcionalidades

¿Se ha podido registrar en nuestro sistema?

- ☐ 0. Sí
- ☐ 1. No lo he probado
- ☐ 2. No

¿Podría insertar una tesis sin su lector?

- ☐ 0. No
- ☐ 1. No sé
- ☐ 2. Sí

¿Si se produce una incidencia sabría con quién contactar?

- ☐ 0. Sí
- ☐ 1. No sé
- ☐ 2. No

Cuando inserta un dato, no se muestra en el sistema. ¿Sabe por qué?

- ☐ 0. Sí
- ☐ 1. No lo he probado
- ☐ 2. No

Cuando edita un dato, no se muestran en el sistema los cambios. ¿Sabe por qué?

- ☐ 0. Sí
- ☐ 1. No lo he probado
- ☐ 2. No

Cuando elimina un dato, no se muestran en el sistema los cambios. ¿Sabe por qué?

- ☐ 0. Sí
- ☐ 1. No lo he probado
- ☐ 2. No

¿Entiende la funcionalidad de nuestra aplicación web?

☐ 0. Sí

☐ 1. No lo he probado

☐ 2. No

*Figura 2: Apartado funcionalidades

Este aspecto contiene preguntas más específicas ya que se pide a los usuarios que naveguen por la aplicación web probando todas las cosas que encuentren.

El método de evaluación en este apartado es similar al apartado de interfaces en el cual contra menor valor sea la suma de todas las respuestas, mejor será la valoración de este aspecto.

Facilidad de uso

Que le ha parecido el procedimiento para registrarse

- ☐ 0. Fácil
- ☐ 1. Medio
- ☐ 2. Difícil

Que le ha parecido el procedimiento para iniciar sesión

- ☐ 0. Fácil
- ☐ 1. Medio
- ☐ 2. Difícil

Que le ha parecido el procedimiento para cerrar sesión

- ☐ 0. Fácil
- ☐ 1. Medio
- ☐ 2. Difícil

Que le ha parecido el procedimiento para buscar una tesis

- ☐ 0. Fácil
- ☐ 1. Medio
- ☐ 2. Difícil

Que le ha parecido el procedimiento para buscar un doctor

- ☐ 0. Fácil
- ☐ 1. Medio
- ☐ 2. Difícil

Que le ha parecido el procedimiento para insertar una tesis

- ☐ 0. Fácil
- ☐ 1. Medio
- ☐ 2. Difícil

Que le ha parecido el procedimiento para insertar un doctor

- ☐ 0. Fácil
- ☐ 1. Medio
- ☐ 2. Difícil

Que le ha parecido el procedimiento para editar una tesis

- ☐ 0. Fácil
- ☐ 1. Medio
- ☐ 2. Difícil

Que le ha parecido el procedimiento para editar un doctor

- ☐ 0. Fácil
- ☐ 1. Medio
- ☐ 2. Difícil

Que le ha parecido el procedimiento para eliminar un dato

- ☐ 0. Fácil
- ☐ 1. Medio
- ☐ 2. Difícil

*Figura 3: Apartado Facilidad de uso

Después de pedir a los usuarios que prueben todas las funcionalidades del sistema se pregunta sobre qué les parecen los procedimientos para llevar a cabo cada caso.

A su vez, comparte el mismo método de evaluación en el que contra menor sea el valor del resultado, más positiva será la valoración de este aspecto.

9.3. Perfil de los encuestados

Para realizar estos cuestionarios, se han intentado buscar perfiles diferentes entre sí para intentar obtener una perspectiva más global del sistema.

En este caso estas son las personas que realizaron el test:

Nombre y apellidos	Edad	Área	Id encuestado
Alejandro Ruiz Lara	23	Ingeniería del Software	1
Cintha Redondo Soto	23	Química	2
Leticia Pérez Jorge	27	Criminología	3

*Tabla 1: Perfiles encuestados

9.4. Resultados del Test

Como se ha comentado anteriormente, los resultados del test se han obtenido asignando un valor numérico a cada respuesta. De este modo según la media obtenida en cada test podemos tener una ponderación de la calidad de cada apartado.

Cuestión	Encuestado 1	Encuestado 2	Encuestado 3	Media
1	0	1	1	1
2	0	0	0	0
3	0	0	2	1
Interfaz	0	1	3	1
4	0	0	0	0
5	0	0	1	0
6	0	1	2	1
7	0	0	1	0
8	0	0	0	0
9	0	0	1	0
10	0	0	0	0
Funcionalidades	0	1	5	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0
16	0	0	1	0
17	0	0	0	0
18	1	1	1	1

19	0	1	1	0
20	0	0	0	0
Facilidad de uso	1	2	3	0

Entrando a valorar los resultados, tenemos que remarcar la diferencia de valoraciones entre alguien del ámbito informático y alguien de otra área totalmente distinta.

Esto nos hace ver que es importante utilizar una variedad de perfiles porque no puedes tener plena firmeza de que el perfil de usuario que acabe accediendo a tu sistema tenga la misma experiencia o conocimiento que otras personas.

Si valoramos los resultados, pasamos a remarcar dos cuestiones las cuales tuvieron un resultado más alarmante.

Ante la cuestión de si sabían que hacer ante una incidencia, algunos usuarios no vieron la pestaña de ayuda por lo que optamos por incluir este detalle en la información de la página principal.

Ante la cuestión de la sencillez de editar una tesis, existió una duda general la cual remarcaban que no podían modificar el lector de una tesis. Según especificaciones del sistema se comprendió que esta funcionalidad no tenía ninguna utilidad y si se creó una tesis relacionada con un doctor erróneamente, debería eliminarse y crear una nueva.



10

Conclusiones

Se finaliza la memoria perteneciente al trabajo de fin de grado “Árbol Genealógico de la Informática”, con un apartado de conclusión en el que se analizarán los objetivos cumplidos respecto a la idea original del proyecto, se comentarán los contratiempos encontrados durante el desarrollo del proyecto y se fijará la proyección futura del proyecto.

10.1. Objetivos cumplidos

Comenzamos con el análisis de los objetivos que hemos conseguido cumplir sobre la especificación inicial del proyecto. A pesar de que hemos seguido una metodología iterativa en la que el registro de funcionalidades se ha visto modificado durante el tiempo se permite hacer una evaluación objetiva del producto final sobre la idea inicial.

Principalmente, se ha conseguido desarrollar una aplicación web completamente funcional en la que hemos podido incluir la totalidad de las funcionalidades que el cliente pedía.

Debido a que se utilizó una metodología ágil en la que el desarrollo del sistema progresó de forma iterativa, el cliente pudo ir comprobando el estado del producto y realizar sugerencias sobre cambios que debían realizarse sobre el sistema.

Es por esto, por lo que se concluye, que el autor del trabajo de fin de grado, en el rol de dirigir y desarrollar un proyecto ha sido capaz de utilizar una mentalidad crítica a la hora de desarrollar el proyecto a pesar de la falta evidente de tiempo para dirigir un proyecto completo sabiendo escoger las tecnologías que mejor se adaptaban a las necesidades del proyecto y elaborando una memoria descriptiva del desarrollo total de producto software.

10.2. Dificultades encontradas

Debido a que en los requisitos iniciales del proyecto se encontraba realizar el desarrollo de gran parte del código en Golang, un lenguaje de programación con un corto periodo de vida y a falta de una gran comunidad detrás como puede ser otros lenguajes como Java o Javascript, este proyecto ya se presuponía de gran complejidad debido a las posibles dificultades que se encontrarían.

A parte de contar con un periodo de tiempo destinado al trabajo de fin de grado destinado a aprender este nuevo lenguaje de programación, la cantidad de contratiempos encontrados a lo largo del desarrollo a provocado que la planificación inicial se viera afectada en gran medida.

Quizás, el problema principal era el de la falta de una comunidad que programe en este lenguaje ya que cuando nos encontrábamos ante un problema en la implementación, apenas encontrábamos información en foros online como puede ser Stackoverflow (www.stackoverflow.com)

Lejos de eso, también nos encontramos con un gran problema a la hora de alojar nuestra aplicación web en un servidor. Ante la negativa de poder alojarlo en los servidores pertenecientes a la facultad de ingeniería informática debido a que no eran compatibles con la tecnología utilizada, se forzó a buscar otra opción viable externa.

Esta tarea también contó con muchos problemas ya que las opciones principales que consistían en plataformas como servicio (Paas) como Google

App Engine o Heroku no fueron factibles debido a la cantidad de esfuerzo que costaba implementarlo (Caso de App Engine) o incompatibilidad (Heroku). También se estudió el caso de utilizar tecnologías como Docker, pero no se disponía de suficiente tiempo como para intentar utilizarlas así que se optó finalmente por una máquina virtual en la nube.

A nivel de proyecto, las dificultades aparecieron principalmente sobre la planificación ya que la indisponibilidad en ciertos periodos de los directores y autor para realizar las reuniones forzaba el proyecto a parar su desarrollo en repetidas ocasiones.

10.3. Líneas futuras

Principalmente, la idea de proyecto iba más allá de la realización del trabajo de fin de grado y se espera que esta aplicación web siga activa en un futuro y se continúe su mantenimiento y soporte.

El problema que podemos encontrar aquí es que actualmente el sistema se encuentra alojado en una IaaS utilizando una versión de prueba gratuita. Si se decidiera continuar con la aplicación web en un futuro se debería buscar una alternativa gratuita o asumir los costes del mantenimiento de alguna forma.

Sobre los planes de expansión de la aplicación, no se definió ninguna extensión futura en la funcionalidad del sistema, si bien, si el sistema cuenta con un tráfico alto, se debería hacer un análisis sobre la conveniencia de seguir utilizando el SGBD actual o migrar la información hacia otro SGBD centrado en la manipulación de grandes flujos de datos



Referencias

-Jaime Echeverri, Miguel Aristizábal, Liliana González. 2013, Reflexiones sobre ingeniería de requisitos y pruebas de software, de <http://ebookcentral.proquest.com/lib/bibliotecauma-ebooks/detail.action?docID=4795310>.

-Martina Seidl, Marion Scholz, Christian Huemer, Gerti Kappel. 2012, UML @ Classroom <https://link.springer.com/book/10.1007%2F978-3-319-12742-2>

-Zhi Jin. 2017, Environment Modeling-Based Requirements Engineering for Software Intensive Systems <https://www--safaribooksonline--com.uma.debiblio.com/library/view/environment-modeling-based-requirements/9780128019573/XHTML/B978012801954218001X/B978012801954218001X.xhtml>



Apéndice

Manual de Usuario

“Árbol genealógico” es una aplicación destinada a representar las relaciones entre doctores mediante los roles representados durante la realización de las tesis doctorales. Además, permitirá ver información sobre los propios doctores y las tesis realizadas de modo que puedas encontrar las tesis en el sistema a través de un buscador.

¿Cómo realizar estas acciones?

Podrás acceder a todas estas funcionalidades a través de la interfaz accesible desde la url (<https://46.101.0.69:8000>). Esta interfaz incluye una guía interna y marcadores que ayudarán a utilizar el sistema correctamente.

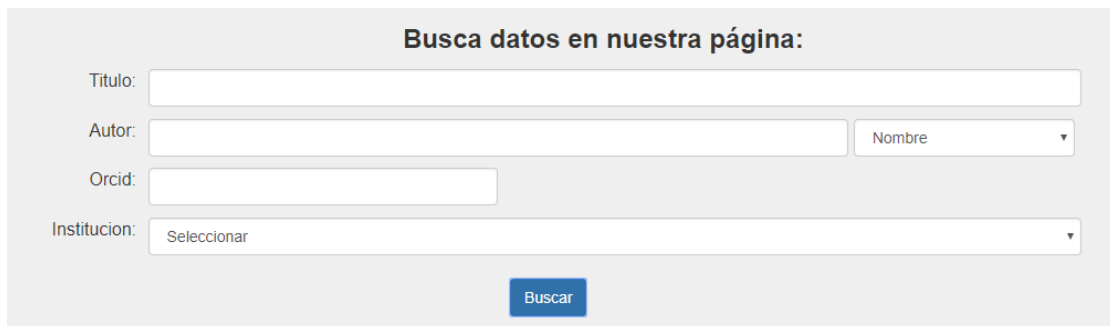
Sin embargo, nuestro sistema restringe el acceso a distintas funcionalidades según el tipo de usuario que utilice nuestra aplicación para mejorar la seguridad y mantenimiento de nuestro sistema.

·Inicio por defecto

La página principal por defecto de nuestro sistema contendrá una pequeña guía del sistema que informará a los usuarios de las funcionalidades disponibles en caso de no estar identificado y de los beneficios de registrarse en nuestro sistema. Desde esta página el usuario podrá acceder al sistema de búsqueda para visualizar datos o a la interfaz de inicio de sesión.

·Búsqueda

En la interfaz de búsqueda nos encontraremos con un formulario en el que podremos insertar los datos que queramos buscar en nuestro sistema.



Busca datos en nuestra página:

Titulo:

Autor: Nombre

Orcid:

Institucion:

Imagen 1: Formulario búsqueda

En este formulario podremos buscar por datos relacionados al autor de la tesis, la propia tesis y la institución donde se realizó la tesis.

Los resultados se mostrarán en una lista en la parte inferior de la interfaz y a través de estos resultados los usuarios podrán acceder a la pestaña de información del doctor o tesis seleccionada.



·Autor: Loli Burgueño Caballero
·Titulo: ON THE QUALITY PROPERTIES OF MODEL TRANSFORMATIONS: PERFORMANCE AND CORRECTNESS
·Autor: Antonio Vallecillo Moreno
·Titulo: Un Modelo de Componentes para el Desarrollo de Aplicaciones Distribuidas
·Autor: Jose Maria Troya Linero
·Titulo: ORDENACION LINEAL DE GRAFOS PARA EL ALMACENAMIENTO DE REGISTROS EN MEMORIAS SECUNDARIAS.

Imagen 2: Resultados búsqueda

En donde si pulsas en el autor, te redireccionará a la visualización de los datos del autor (Imagen 3) y si pulsas en la tesis te redireccionará a la interfaz de visualización de la tesis (Imagen 4)

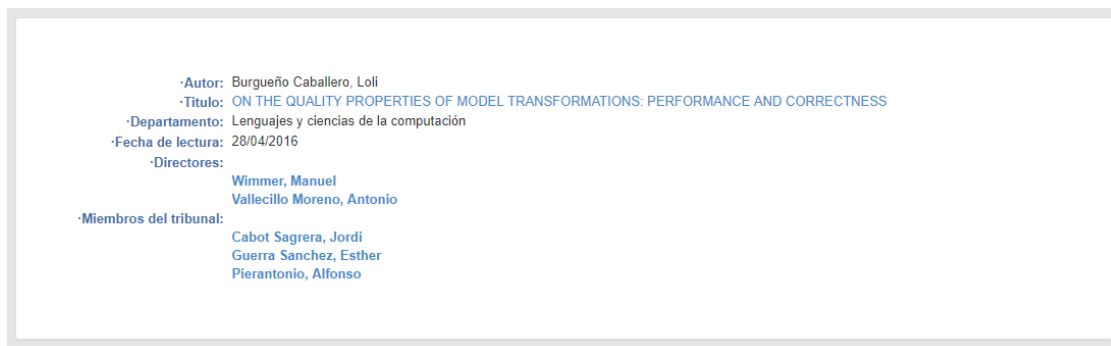


Imagen 3: Visualización doctor

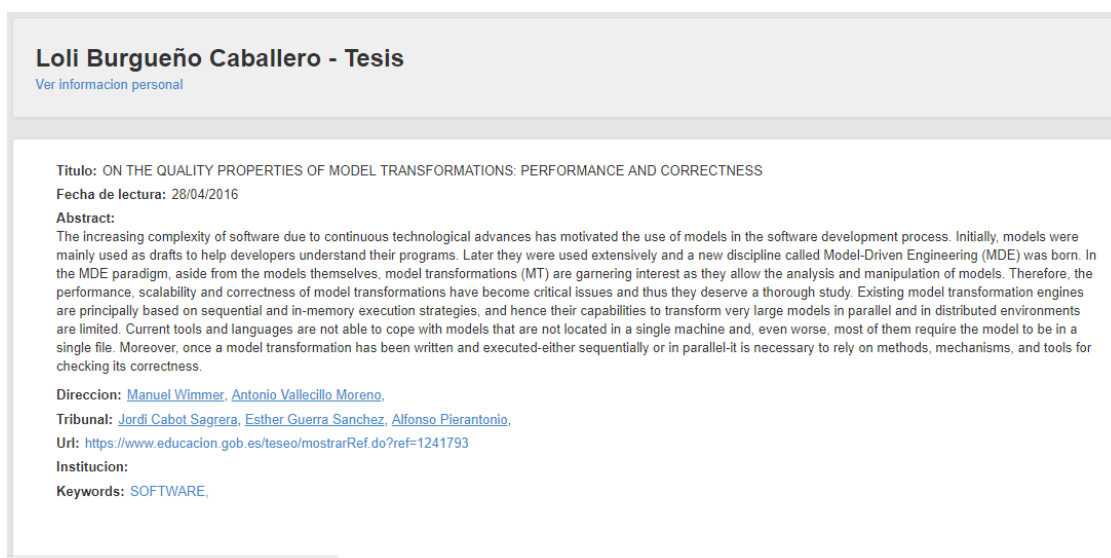


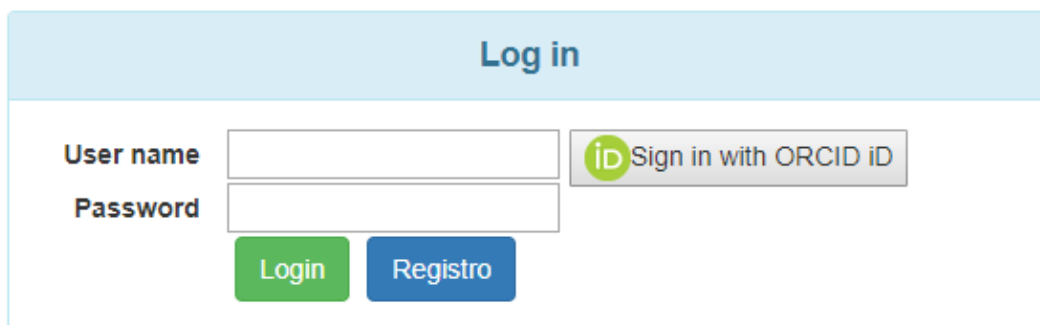
Imagen 4: visualización tesis

•Contacto y ayuda

Otra funcionalidad accesible para todos los usuarios es la de ayuda y contacto la cual dispone de una interfaz accesible mediante el botón ayuda y contacto en la parte derecha del menú superior los usuarios dispondrán de un conjunto de dudas comunes y un formulario de contacto para poder informar de anomalías o pedir que se elimine algún dato.

•Log in

Esta interfaz, cuenta con un panel situado en el centro de la pantalla en el que los usuarios pueden insertar sus datos de acceso para iniciar sesión o seleccionar la opción de registrarse e insertar sus datos de registro.



The image shows a 'Log in' panel with a light blue header. Below the header, there are two input fields: 'User name' and 'Password'. To the right of the 'User name' field is a button labeled 'Sign in with ORCID iD' with a green 'iD' icon. Below the 'User name' and 'Password' fields are two buttons: a green 'Login' button and a blue 'Registro' button.

Imagen 5: panel Log-in

Al iniciar sesión en esta página se redirigirá automáticamente a la página principal pero ahora podrá visualizar un cambio en el menú superior y es que al haberse identificado en nuestro sistema cuenta con otras funcionalidades disponibles.

•Menú usuario identificado

Cuando pasas a ser un usuario identificado en el menú superior se añadirán dos nuevas pestañas de navegación:

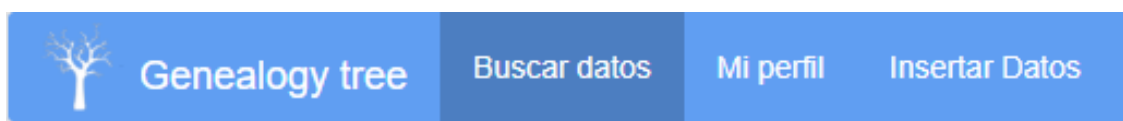


Imagen 6: Menú usuario identificado

Mi perfil: Redirección a la visualización de datos del propio usuario

Insertar Datos: Primera interfaz significativa para los roles identificados los cuales podrán insertar nuevos datos en el sistema a través de un formulario en el que desde un primer momento podrás insertar las relaciones con doctores o con la institución.

En donde si queremos insertar un doctor y pulsamos una de las cajas de textos pertenecientes a ellos nos permite buscarlo a través de una nueva ventana flotante en la cual, si seleccionas uno de los resultados en la tabla, se carga en el campo de texto.

Una ventana similar se abrirá si queremos insertar la institución en la que también podremos buscar y seleccionar la institución que queremos cargar

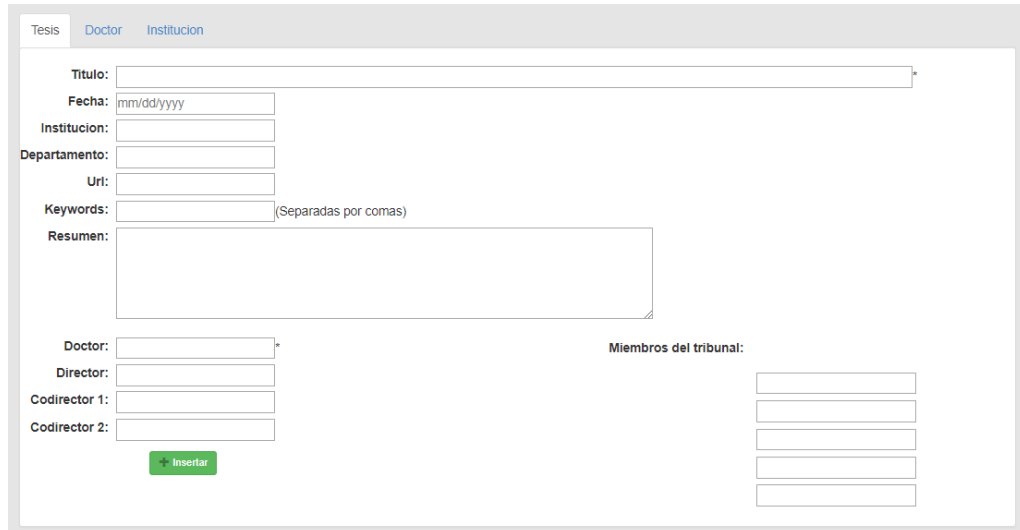


Imagen 7: Interfaz insertar datos

Una vez introducidos los datos, aparecerá un mensaje de confirmación de inserción y de aviso que los datos antes de aparecer en nuestro sistema serán confirmados por un administrador.

•Editar datos

Otra funcionalidad adquirida con el registro es la de poder editar datos ya existentes en el sistema.

En las pestañas de visualización de datos mostradas anteriormente (Imágenes 3 y 4) ahora aparecerá un botón junto con el texto “Editar información”. Al pulsar este botón nos redireccionará a una interfaz muy similar a la de inserción de datos en la que se cargarán los campos pertinentes con los datos existentes de la tesis o doctor seleccionado y podremos realizar las modificaciones que creamos pertinentes y enviarlas

Tesis
Doctor

Título: ON THE QUALITY PROPERTIES OF MODEL TRANSFORMATIONS: PERFORMANCE AND CORRECTNESS
Fecha: 28/04/2016
Institucion:
Departamento:
Url: https://www.educacion.gob.e
Resumen: The increasing complexity of software due to continuous technological advances has motivated the use of models in the software development process. Initially, models were mainly used as drafts to help developers understand their programs. Later they were used extensively and a new discipline called Model-Driven Engineering (MDE) was born. In the MDE paradigm, aside from the models themselves, model transformations (MT) are garnering
Lector: Loli Burgueño Caballero
Director: Manuel Wimmer
Codirector 1: Antonio Vallecillo Moreno
Codirector 2:
Jurado: Jordi Cabot Sagrera
Jurado: Esther Guerra Sanchez
Jurado: Alfonso Pierantonio
Jurado:
Jurado:

Imagen 8: Interfaz edición de datos

Los cambios realizados, a su vez, deberán ser aceptados por un administrador antes de que se vean modificados en el sistema.

Normas aplicadas

Para el desarrollo de este proyecto, y en específico, esta memoria del proyecto se han seguido dos normas estandarizadas:

-UNE – 157801: Norma desarrollada por la asociación de normalización española (AENOR), define los criterios generales para la elaboración de proyectos de sistemas de información. En esta norma se especifica los entregables que se deberían presentar en caso de desarrollar un proyecto software. Sin embargo, este proyecto pertenece a un trabajo de fin de grado por lo que se creyó conveniente no incluir algunos documentos como los análisis de presupuesto ya que se prefería invertir el tiempo en un mejor desarrollo del sistema.

-Normas de referencia: En este caso se hemos seguido las normas de referencias establecidas por la American Psychological Association (APA).